An investigation into the use of Multi-Agent Systems in marine simulator instructor stations

James Nicholas John Moon

A thesis submitted in partial fulfilment of the requirements of the

University of Glamorgan/Prifysgol Morgannwg

for the degree of Doctor of Philosophy

July 1997

MARINES

Table of Contents

1. Table of Contents

2. Table of Figures

1.

2. Declaration

This is to certify that neither this thesis or any part of it has been presented or is being currently submitted for any degree other than the degree of Doctor of Philosophy at the University of Glamorgan.

Candidate ______________________________

3.

4.

5. Certificate of Research

This is to certify that except where specific reference is made, the work presented in this thesis is the result of the investigation undertaken by the candidate.

Candidate ______________________________

Director of Studies ______________________

Acknowledgements

I would like to take this opportunity to extend my sincere thanks to my Director of Studies, Dr. Douglas Tudhope, and my Supervisors, Professor Christopher Jones and Dr. Paul Beynon-Davies. Throughout the project they have guided the work in an enthusiastic and confident manner, for which I am grateful.

My grateful appreciation is also extended to all those who have given their time to provide advice, evaluate developments and discuss the progress of the work. In particular, I would like to thank Mr. Pat Cleary, Captain Alan Frost, Dr. Ian McCallum, Captain Robert Rayner, Mrs. Geneen Stubbs and Captain Ken Waters, all of whom have provided astute judgements that have helped to focus my thoughts.

I would also like to acknowledge Mr. Peter Hodson and the University staff who have supported the development of research within the Department of Computer Studies and the University of Glamorgan.

I would especially like to acknowledge the support of my wife, Hui-Jung, who's continuous good humoured encouragement has been an inspiration throughout this project.

Finally, I would like to take this opportunity to remember Captain Alan Donald, who passed away shortly before the publication of this thesis. Captain Donald assisted in the design of a number of exercises based upon his extensive experience as a master mariner. He was particularly thorough when he assisted in the evaluation of MARINES and his shrewd opinions supported some important findings.

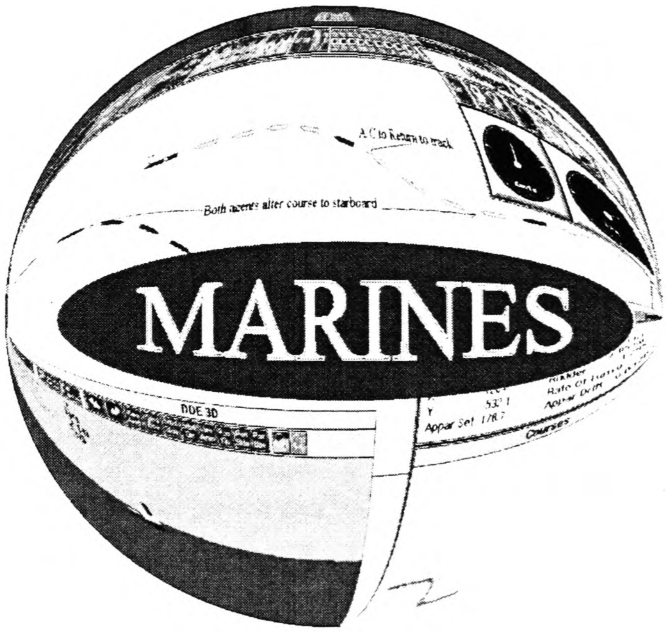
Abstract

This thesis documents an investigation into the automatic provision of reasonably realistic motion for the computer generated target ships in a marine simulator. The thesis explores: automatic collision avoidance between the target ships; automatic track keeping for the target ships; the use of sea stabilised and land stabilised motion for the target ships; some issues of software fault tolerance in marine simulators; message frameworks for use in a Multi-Agent System (MAS) simulation; the opportunity to provide different manoeuvring characteristics for different target ships; and the use of autonomous agents to control the target ships.

A software system has been developed to facilitate this research. Entitled "A **M**ulti-**A**gent **R**ealm for **I**nvestigating **N**avigators' **E**ducational **S**imulators" (MARINES), the software is a MAS providing much of the functionality of a marine simulator instructor station; basic functions are encapsulated into the instructor environment and additional features are provided by processes that connect to the environment using Dynamic Data Exchange. The processes can also connect to each other and, in MARINES, co-operate to navigate the ships. These co-operative, autonomous processes are the agents that together form a MAS. A simple 3D view is also connected, enabling the view from the bridge of a specific target ship to be assessed. The MARINES software is written using C++ to run under Microsoft Windows v3.1. Therefore, the processes multi-task co-operatively.

In MARINES each target ship can be made to perform in an individual manner; manoeuvring and performance characteristics can be customised to simulate a specific ship type. Additionally, the agents performing collision avoidance can be given rule sets that interpret the International Regulations for the Prevention of Collisions at Sea in subtly different ways, and the track-keeping agents can have different beliefs about the manoeuvring capabilities of the vessels they control. Automatic collision avoidance and track keeping is performed for two-ship situations even when the set and drift of a current is introduced. A comparison is made with the tracks of land stabilised targets. This shows how aspect, course and speed are affected by ignoring the effects of the current, and demonstrates the need for an accurate simulation.

Chapter One



Introduction

1. Introduction

1.1 Overview

This chapter gives a brief rationale for the project, sets out the aims and objectives and discusses the layout and formatting of the document.

Sections 1.2 and 1.3 give very brief descriptions of the areas of marine simulation and Multi-Agent Systems (MAS) that are discussed in this thesis. These provide a basic introduction to set the scene for the project. For a more detailed description of simulation and Multi-Agent Systems chapters 2 and 3 should be consulted.

In section 1.4 the rationale as to why the project has been undertaken is discussed, the rationale is expanded upon in chapter 4. The rationale, together with the objectives of the project, given in section 1.5 and the sequence of work in section 1.6 show the route taken to the development and conclusions of the thesis.

The layout of the content of the document is summarised in section 1.7. The primary topic for each chapter and the positions of the major sub-topics are listed.

Finally a few of the main points are summarised in section 1.8.

1.2 Marine simulation

Marine radar and bridge simulators are training aids for navigation students and pilots. They are also a tool for designing harbours and port approaches. Another common use is the reconstruction and analysis of marine disasters. Such a simulator consists of one or more facsimiles of a ship's bridge and an instructor station.

The instructor station normally consists of a computer terminal showing information about the ships in the simulation. For example, course, speed, position etc. The display will also normally contain a map of the simulated area. Facsimiles of VHF sets and internal communications panels are also provided. Using the instructor station, exercises are designed, set up and supervised by the simulation instructor. The instructor also controls the computer generated ships and responds to any radio communications at run-time.

A simulator is particularly good at providing scenarios:

- that do not yet exist, for example, a new port facility or ship type;
- that would prove too dangerous to create in the real world, for example, a previous collision;
- that provide basic collision avoidance, for example, radar plotting practice.

This gives the students at least a little experience of a danger that may otherwise present itself in an all too real manner.

A more complete description of simulation and simulators is given in chapter 2.

1.3 Multi-Agent Systems (MAS)

The Multi-Agent Systems discussed in this thesis form a relatively new branch of Distributed Artificial Intelligence (DAI) (O'Hare and Jennings 1996). A MAS tries to produce complex effects through the use of a large number of relatively simple agents. Each agent is able to act autonomously to achieve goals, without continuous user intervention.

Growth in MAS research has been rapid and dynamic since the late nineteen eighties. The major concentration of this research has been in MAS theories, architectures, languages and test beds. While the potential of this MAS research has been widely accepted, until recently, few real world MAS applications have been built. One commercial application area that has been successful is the creation of MASs that perform monitoring processes, such as factory shop floor control systems.

Autonomous, intelligent agents have also performed information retrieval on behalf of software users. Once again, this area has seen an explosion in interest. Much of this type of research may be seen as falling within the Information Systems (IS) field. However, the standardisation of agent interfaces and co-operation between agents is of growing importance and MAS researchers are now working on these problems.

In diverse areas of computing agent technology is now becoming de rigueur. More and more companies are citing agent capabilities for their software; it seems that agent technology "*will soon be ubiquitous and essential*" (Gurton 1996). Some of these packages are little more than the latest version of an existing software package and contain limited agent technology. However, some of the benefits of agents, such as flexibility, autonomous action and robustness appear genuine.

Agents and Multi-Agent Systems are discussed further in chapter 3.

1.4 Rationale for the project

There appears to be much to be gained from the application of agents to simulation, particularly in providing a robust, flexible system with reasonably realistic target ship motion generated automatically.

Chen (1992) researched the use of marine simulators in Taiwan ROC and found that: many of the simulator instructors were specialist navigators but not specialists in computer simulation and were intimidated by the simulators; the simulators were difficult to use; run-time errors occurred that terminated the simulations, further intimidating the instructors; the simulations also behaved in ways that were not realistic.

Guicharrousse(1990) found that simulations were unable to reproduce the complexity of the real world. Because of this the students became good at recognising what to do during exercises. They would know how to respond to a given situation because the computer behind the simulation was only able to respond as it had been programmed to do. When faced with the same problem in the real world the response might be less easy to determine. While MARINES does not attempt to fully reproduce the real world effects, the responses of the target ships are intended to be more realistic and less predictable; small changes in the simulated environment causing dynamic changes to the response of the agents.

The committee on ship-bridge simulation training discusses the importance of accurate ship motion and finds that "*The ability of a simulator to replicate closely a ship's maneuvering trajectory is a strong measure of the usefulness and value of the simulator for training and licensing*" (Meurn 1996). Furthermore, this supported a recommendation that "*The U.S. department of Transportation should develop standards for the simulation of ship maneuvering*". From the context of the document it is inferred that these recommendations are intended largely for 'own[1]' ships. However, one of the intentions of this research is to show that computer generated ships should also manoeuvre realistically. This is particularly true when environmental effects such as the set and drift of a current are applied. A rationale for this is given in chapter 4 and exercises are performed to demonstrate the effects in chapter 8.

It is possible that intelligent agents can provide some assistance to the instructor. At the same time, a MAS should create a more flexible, dynamic simulation. The partial control of the computer generated target ships, by intelligent agents, may provide a step in this direction. The agents do not have to be identical; agents with individual characteristics can be created. This appears to offer potential for a simulator that provides at least some of the complexity of change found in the real world. Simulators usually contain large complex software systems. It is generally accepted that, using current development and testing strategies (Pressman 1992), such systems will contain some errors. The lack of repeatability in continuous simulations makes it even more difficult to eliminate errors by dynamic testing. The asynchronous nature of the problem, hardware in-the-loop and interrupt driven events also increase the likelihood of an error terminating the simulation. Additionally, simulator exercises can run for several hours. If an error occurs that ends the simulation in an uncontrolled manner a considerable amount of effort may be wasted. It may also be impossible to obtain the same results in a re-run of the same exercise; apart from the non-repeatability, the students will have already gained experience and may choose a new strategy. Furthermore, the students and the instructor will lose confidence in the system. The use of a full-mission bridge simulator is also very costly, certainly hundreds and, for many systems thousands, of pounds per hour. Intensive simulator courses only allow the students a few hours bridge time each. Therefore, loss of exercise time is to be avoided as far as possible. If the simulation can continue, transparently to the student, after some software errors, then MASs may very well offer a useful alternative to conventional simulation architectures.

[1] The ships navigated by the students.

1.5 Project Aims and Objectives

The primary aim of this project is to critically evaluate the benefits of using MASs in marine simulation.

More specifically, one aim is the critical examination of the ability of a MAS to provide reasonably realistic target ship motion, through interaction between a number of relatively simple agent processes. It is intended to include ship models that provide generic manoeuvring characteristics for different ship types and are affected by some environmental effects. In assessing the ship motion, a further aim is to evaluate the agents' ability to manoeuvre these models in a reasonably realistic manner. Thereby, determining whether MAS agents are able to provide automatic track-keeping and automatic inter-ship collision avoidance in a dynamically changing environment, avoiding other vessels and maintaining the desired tracks as appropriate.

The final aim is to analyse the ability of a MAS to provide a robust, flexible platform for a marine simulation. The intention is to evaluate the ability of a MAS to exhibit tolerance to certain software errors and determine a suitable framework for a marine MAS.

The following key objectives have been identified:

- The creation of a MAS system architecture; an architecture that provides sufficient communication bandwidth and adequate performance for the simulator to run at a real world time rate. This depends upon several factors: the design of the MAS architecture; whether the messages are processed immediately or stored for future processing; the priority of the messages; and the design of the message language.
- The creation of agents to perform tasks in the simulation and the critical evaluation of the ability of the agents to adapt, provide adequate performance and accurate behaviour under both planned and reactive conditions. The creation of a general interface to allow the agents to communicate with each other, retrieve information from the environment and control the ships.

- Critical evaluation as to whether the manoeuvring of the target vessels closely resembles behaviour found in the real word, including evaluation of whether the motion of the computer generated target ships can be affected in a natural manner by changes in the environmental conditions and characteristics of the agents.
- The evaluation of whether an instructor can be assisted in controlling the exercises by intelligent agents. Agents may be able to monitor the situation and alert the instructor to dangerous situations that are developing, permitting the instructor to concentrate on the local area around the own ship.
- The determination of whether the simulation can continue to run after a run-time error in part of the system that is not critical. The error may be contained in a part of the software that provides a feature that is useful to have but is not essential. If the damage to the simulation could be restricted to this non-critical area then a more controlled approach to the repair and testing could be taken. The failure of one of these agents would reduce the functionality slightly but not prevent the simulation from being completed successfully.
- The evaluation of whether a MAS can provide a flexible, configurable platform for an instructor station. This requires investigation into the ability to start and stop agents, connect different agents to the environment and for the provision of agents with individual characteristics.

1.6 Work Undertaken

There follows a list of the main areas of work that have been undertaken. Although the topics are ordered approximately on start date, no schedule has been implied. Research and development has generally progressed on three parallel routes: researching MAS literature; researching marine collision avoidance and simulation; and developing the MARINES system. The topics below are those that have been undertaken as research has dictated. This differs from the original plan of work in a number of ways, the most significant being the introduction of the study of environmental conditions upon the simulation and less complete coverage of collision avoidance. In particular collision avoidance situations involving several ships have not been addressed, however, several two ship situations can occur in the same scenario. The main topics that have been covered are:

- A study of the theories, architectures and languages used in previous MASs.
- The determination of a suitable MAS architectural framework for a continuous simulation.
- A study previous work on simulation.
- A study previous work on marine collision avoidance.
- The selection of an internal architecture for each agent in the simulation.
- The production a marine simulator instructor station as a central environment for a MAS.
- The creation of object oriented mathematical ship models to be controlled by intelligent agents.
- The creation of intelligent agents to control the mathematical ship models. Each agent is to connect to the instructor station environment. Some of the agents also connect to each other.
- A critical study of the communication between the agents controlling the computer generated target ships.
- An evaluation of the capability of intelligent agents to provide collision avoidance between the computer generated target ships.
- An evaluation of the capability of intelligent agents to provide track keeping for the computer generated target ships.
- An investigation into the use of intelligent agents to counteract the set and drift of a current for the computer generated target ships.

- A critical evaluation of the motion of computer generated target ships under the automatic control of intelligent agents and determination as to whether the result provides a reasonably realistic target motion and 'aspect'[2] of approaching vessels.
- The critical consideration of the effects of introducing an agent that has a misconception about the rules of the world it is inhabiting.
- A study of the effect of changing environmental conditions upon the developing scenario.
- An investigation into whether simulation software can be produced that permits the simulation to continue running with reduced functionality after a run-time error; if the error occurs in a part of the software that is not critical to the simulation.
- The views of some simulation experts about the qualitative value of the instructor station MAS that has been produced.

This work has led to the creation of a Multi-Agent System for a marine simulator instructor station. This includes the selection of a message framework for use in a Multi-Agent System (MAS) simulation and the use of autonomous agents to control the target ships. The simulation provides reasonably realistic motion for the computer generated target ships in a marine simulator. This includes the combination of automatic collision avoidance between the target ships with automatic track keeping. This is done for a number of different ship models with generic manoeuvring characteristics for different ship types. Unlike earlier simulations incorporating collision avoidance the target ships are affected by the set and drift of the current. Additionally, the agents performing collision avoidance can be given rule sets that interpret the International Regulations for the Prevention of Collisions at Sea in subtly different ways, and the track-keeping agents can have different beliefs about the manoeuvring capabilities of the vessels they are controlling. The environmental effects

[2] The aspect of a vessel describes the way that an approaching ship appears to an observer. It is actually a measure of the angle that the observer subtends relative to the heading line of the approaching ship.

and target ship course and speed can be adjusted at run-time, the agents adapting dynamically to the changes.

1.7 Chapter Content and Layout

Chapter 2 gives a brief description of discrete event simulation and continuous simulation. Then the use of simulators for training is discussed. Some aircraft and car simulations have similar layouts and objectives, therefore, examples of these are described briefly. After this marine simulation is described, first in general and then a more detailed look at bridge simulators. Finally, traffic management, track keeping and marine collision avoidance are discussed.

Chapter 3 describes Multi-Agent Systems (MAS). Section 3.2 discusses the term Multi-Agent Systems and some of the key terms used in the discipline: intelligence; autonomy; etc. Section 3.3 describes MAS architectures, differentiating between system architectures and internal agent architectures. In section 3.4 agent languages and tool kits are described and section 3.5 describes some previous agent based simulation environments. Section 3.6 suggests some factors that should be considered when starting a new MAS development.

Chapter 4 discusses some developments that would be desirable in a marine simulation, expanding upon the rationale for the project as discussed earlier in this chapter.

Chapter 5 provides a description of MARINES at a system level. The chapter begins with a brief conceptual overview of the system. Essentially, the system comprises a marine simulator instructor station which agents are able to connect to, providing additional services. Currently, intelligent agents provide track keeping and collision avoidance for the computer generated target ships. The conceptual view of MARINES is expanded upon, the environment, the ships, the inter-agent connectivity, etc. Next, there is a short discussion about the choices of operating system, programming language, and methodologies used in the construction of MARINES. A major part of the chapter is then set aside for a description of the message passing mechanism; this forms the communications framework upon which MARINES relies to link the processes together.

Having described the system framework in chapter 5, chapter 6 takes the central environment around which the agents are clustered and details the internal architecture. This is the main instructor process; it provides ships, environmental conditions, the ability to start and end the simulation, etc. Without the agents, or any other processes, it offers a sub-set of the functions found in a typical instructor station. Once again, the description begins with the conceptual design. This is followed by a high level portrayal of the software design. Each component is described: the mathematical ship models; transmitters and receivers; etc. After this the instructor environment interface is explained.

Chapter 7 describes the internal architecture of the agents. A high level view of the actual software architecture for the collision avoidance agent is given and each major component explained. Next, the high level design of the track-keeping agent is given and the components that differ from the collision avoidance agent are discussed. After this, there is a comparison between the similarities and differences from other agent systems described in chapter 3.

Chapter 8 describes the exercises that have been performed to assess the accuracy of the MARINES components and whether the system achieves the project objectives. The start of the chapter describes the experiments that determine the initial manoeuvring characteristics of the ships. Then the tuning of the models is explained and the final manoeuvring characteristics are tabulated. Following this, tests are described about the track-keeping capabilities of the agents. There is then a description of the tuning of the auto pilot settings. The track-keeping properties of three different ship models under agent control are then compared. Collision avoidance trials for a number of simple, scenarios involving two ships came next on the agenda. Once the basic avoidance mechanism had been verified for a limited number of scenarios then the effects of changing the set and drift of the current were investigated. This attempts to, at least partially, answer one of the project objectives for reasonably realistic motion under changing conditions. Next the effects of a run-time error are tested and the results given. The results of the experiments are then considered. In particular, the effects of current and collision avoidance on the agents ability to perform track keeping. The loss of one process and reconfiguration of the system is also considered.

Chapter 9 describes the evaluation of the system with expert marine simulation users. The interviews with the expert users are briefly discussed. These interviewees also assisted by filling out a questionnaire; an example of this is given in Appendix C. The responses to the questionnaires are examined and the results considered. The experimental results and the results of the questionnaire are compared. Finally the overall effect of the MARINES instructor station technology is considered. The strengths and the weaknesses of the system are discussed, followed by a discussion on some of the considerations for a commercial implementation.

In chapter 10 the main findings are re-stated, including the motion of the ships and the issues of robustness. Some recommendations about the use of MAS are then made. The MARINES project has commenced the work in the area of MAS in simulation, chapter 10 also suggests future directions the research might take both in the short term and the longer term. Finally, the conclusions of the project are re-stated.

Appendix A contains a brief explanation of the marine collision avoidance and track keeping problems. Appendix B discusses the construction of the collision avoidance rule compiler. The questionnaire that was used as a part of the semi-structured interviews is contained in Appendix C. Brief explanations for a few of the more specialised terms and acronyms are given in the glossary in Appendix D. The cited references are contained in appendix E.

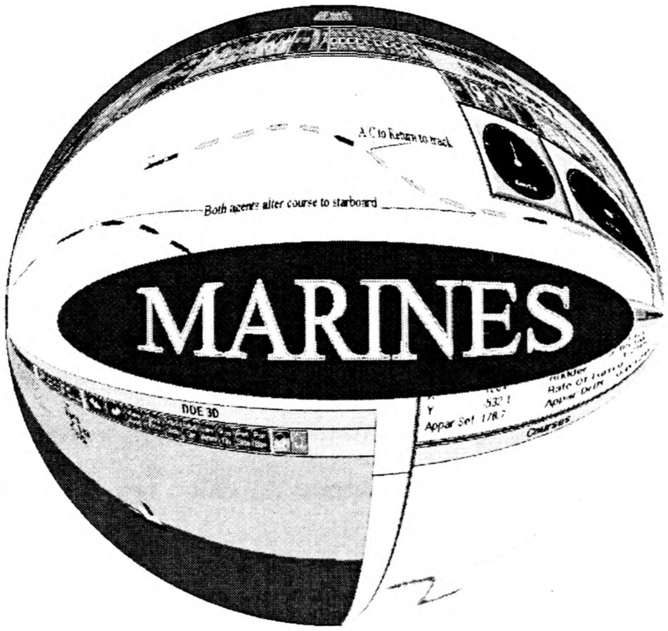
1.8 Summary

This chapter provides an introduction to the content of the document, the reasons for the project and the project objectives.

The areas of marine simulation and Multi-Agent Systems are quite disparate. Therefore, very simple descriptions of the disciplines have been given. The descriptions should help to set the scene and will be expanded in chapters 2 and 3 .

However, the most important aspects of this chapter are the rationale for the project and the objectives of the research. Some of the strengths attributed to a MAS are robustness, flexibility and the ability to create complex, dynamic worlds using a number of relatively simple processes. Intelligent agents have also been attributed with the ability to assist the user of software packages. MASs apparently offer potential *solutions to two criticisms of current simulator technology: some assistance for the part-time or novice instructor; and a move towards real-world complexity and dynamics in a simulation. Essentially, this project is an investigation into the creation of a marine simulator instructor station using a Multi-Agent System (MAS) architecture and an evaluation of the instructor station produced. This thesis provides a commentary on the project and a discussion on the advantages and challenges encountered in the creation and use of such a system. A brief discussion of the content of the chapters has also been given.*

Chapter Two



Simulation and Traffic Management

2. Simulation and Traffic Management

2.1 Overview

This chapter discusses simulation describing discrete and continuous simulations in general in section 2.2. Some examples of training simulators used in disciplines other than marine training are given. A selection of marine simulations are then described to give an impression of the wide use of simulation for marine applications. This is followed by a more detailed section on marine training simulators, ranging from desktop trainers to full mission bridge simulations. This section is particularly relevant to the MARINES project although some principles are applicable to a wide variety of simulations.

Section 2.4 discusses marine traffic management and collision avoidance, it gives a brief description of the two topics and then describes previous work on automatic collision avoidance in more detail. There is then a short description of track-keeping for computer simulated ship models and fuzzy logic auto-pilots. The final section discusses the future potential of automatic ship navigation and brings together all the traffic management and collision avoidance research considered.

2.2 Simulation

Simulation is widely used and no survey can hope to cover every aspect. Computer simulation is broadly split into two categories, discrete event simulation and continuous simulation. As the title suggests, discrete event simulations rely on either manually generated or computer generated events occurring, whereas, continuous simulations try to create realistic models of the actual components that generate the impression of a naturally changing environment. Each category is suited to particular types of problem. Discrete event simulation is often used, where statistical information is available, to generate 'what if' scenarios. These simulations normally run as fast as possible, going straight to the next task as soon as the current task is completed, and the passage of processor time may not be linearly related to the passage of world time. On the other hand, continuous simulation is suited to modelling the passage of time, as in the real world.

2.2.1 Discrete event simulation

"Discrete event simulation works by articulating models of system components only when specified events occur" (Macfadzean et al. 1995). This type of simulation is particularly good for modelling the flow of objects through a number of local bottlenecks. A factory production line is typical of this type of problem and discrete event simulations are often used to optimise the flow of components along the production line; the arrival and departure of components act as triggers to keep the simulation operating. Techniques like Petri nets (Brennen et al. 1995; Ilie et al. 1995) and queuing theory have been successfully used in the design of experiments for these systems and can provide particularly good solutions when a steady state can be achieved. However, this global control can be inflexible and slow to respond to changes both in production rates and equipment configuration changes (Van Dyke Parunak 1996).

Discrete event simulation is also useful to simulate breakdowns based upon historical information. For example, if it is known how many containers can be loaded by one container crane in an hour, discrete event simulation can be used to model what will happen if a crane breaks down. The simulation can help in the estimation of how long it will be before all the available dock space is used to store the containers, or whether the flow of trucks from the depot should be reduced if the crane is expected to have four hours of down time, etc.

These types of simulations generally operate on a large number of instances of simple algorithms. For example, if a loaded container lorry arrives then add it to the shortest queue at the loading dock. This simplicity means that even relatively complex discrete event simulations can often run extremely fast, simulating many hours in only a few minutes of processor time.

2.2.2 Continuous Simulations

Continuous simulations often use Euler integration or Runge Kutta predictor methods to mathematically model movement (McCallum 1980; Pourzanjani 1990a). In general, continuous simulations do not produce precisely repeatable results; tiny changes in timings alter the accelerations and these accumulate to create a different, although similar, outcome from the same initial conditions. Therefore, this type of simulation is ideal for modelling motion in the real world. Unfortunately, for the same reason it is not possible to produce guaranteed results from the same initial conditions. This has two implications, the first is that the same experiment, run several times, may produce mildly conflicting results. The second is that no amount of testing will ensure that the system is 100% reliable, even for the same experiment. Therefore, the debugging of the software becomes difficult and even more emphasis must be placed on good design and re-use of proven components.

Models of each element in a continuous simulation are usually more sophisticated than in a discrete event simulation. The equations used to produce accurate motion may contain a large number of terms and have to be updated frequently. The rate at which the simulation can run is usually limited by the elapsed simulation time for an iteration, this is called delta time or 'dt'. A continuous simulation is usually updated as frequently as possible, therefore, as the simulation rate is increased so 'dt' becomes larger. If 'dt' is too large the models can become inaccurate and in extreme cases run out of control. Several factors are involved including: the speed of the processor and presence or absence of a math co-processor; the complexity of the models; the number of models on each machine; the frequency with which the system clock updates on the computer; the language used; the mathematical precision required and whether the models are based on heuristics or Newtonian motion equations; most are based upon a mixture of the two. For these reasons it is unlikely that such a simulation will produce meaningful results if the world time simulated is a multiple of more than a few times the processor time. Even relatively simple models become unstable at higher rates. For instance, a standard AT-Compatible updates the user clock 18.3 times a second, using Euler integration the accelerations of a ship, particularly in roll, yaw and pitch, may become unnatural as 'dt' approaches one second (Pourzanjani 1990b), therefore, even a simple Euler model will not operate more than about 15 times faster than World Time. This can be overcome by more sophisticated integration methods such as the Runge Kutta Predictor method. However, these are usually also more demanding of the computer system. A further possibility is to manipulate the system clock or provide a timer with a higher frequency. Although usually the other constraints will prevent high iteration times being achieved.

Therefore, continuous simulation is largely used for real-time simulators, where the actual environment is continually changing and affected by several independent factors. All these factors make it extremely difficult, if not impossible, to precisely predict what will happen in five minutes. For example, it is hard to determine the position and speed of a ship in changing weather and tidal conditions, or the movement and shape of a cloud. However, by modelling the causes, rather than predicting the effects, accurate simulations can be created.

2.2.3 Combining Discrete Event and Continuous Simulations

Wildberger (Wildberger 1995) describes methods that may bring discrete event and continuous simulations together. Early simulations were either analogue or digital and Wildberger describes them as analogous to continuous and discrete simulation, respectively. However, it is Wildberger's description of query driven simulation that has more relevance to MARINES. Query driven simulation permits the derivation of information that is not explicitly stored. This is done by running a short continuous simulation. Heuristic rules may be used to infer results from information that may be elicited or developed at run time. Therefore, the discrete event may start a temporary continuous simulation. The agents in MARINES operate through the use of discrete queries. These queries generate short bursts of continuous planning in the agents, followed by periods of waiting for the next event.

2.3 Training Simulators

Many training simulators attempt to re-create real scenarios. Such simulators are valuable when the real equipment is: comparatively expensive; hazardous to use; impossible or inconvenient to use in training. Some other factors that may influence the selection of such training aids are: shortage of training instructors; repeatability of experiments and the ability to simulate many different tableaux, even fictional or future scenarios. This type of simulator has to respond in real-time to the movement of controls. Analogue controls and gauges, and even hydraulic motion platforms, are used to stimulate the senses. Timely solutions (Reddy and Moon 1995) are required to provide these deterministic responses, and thus, efficient algorithms, mathematically adept computer languages and efficient hardware are necessary. An example of an efficient development tool kit would be 'Performer' and 'Multigen' on Silicon Graphics work stations. 'Performer' is a 3D simulation development tool provided by Silicon Graphics. 'Multigen' is a 3D data base editor; together, they can be used to provide rapid development of complete 3D simulations.

The International Convention on Standards of Training, Certification and Watchkeeping for Seafarer's (STCW 1995) developed standards that are to apply to marine training simulators. Cross (1996) states that *"The revision of STCW has formalized what has been the practice in many of the European countries...to assess competence by means of simulators"*. Furthermore, remission from required periods of sea service is being suggested; Cross (1996) continues *"it is agreed that some remission should exist as a further incentive to promote simulator learning"*.

2.3.1 Aircraft simulation

2.3.1.1 Cockpit Simulators

Military and commercial pilot training has used 3D visualisation for many years. These 3D systems employ the latest technology to create highly realistic scenes. The Rediffusion CT5A (Rediffusion 1982) is an early example of a sophisticated 3D simulator visual system, with a claimed capability of displaying up to 1,000,000 pixels per channel in real time on up to eight channels. The system includes features such as smooth shading, sun shading, colour blending and transparency, smoke, dust and rotor downwash for helicopter simulations.

Parallel and pipelined architectures have been developed by companies such as Evans and Sutherland (Rediffusion 1982), Silicon Graphics and Dupont for rendering ever more sophisticated scenes including filled, shaded and shadowed landscapes in real time.

The effect of sound is also being experimented with[1]. Vital time is lost through pilot distraction and the need to cancel alarms. Using special directional head sets, the

[1] This work was described by a guest speaker at SCSC '95.

warnings attract the attention of the pilot towards the danger, not towards a fixed alarm panel. The pilots' attention is immediately centred on the danger.

2.3.1.2 Air Traffic Control Simulations

Real air traffic control equipment incorporates digital computers and raster display technology and is often housed in darkened enclosed spaces. The restricted nature of the real world equipment make it possible to create an air traffic control simulator that is highly realistic.

Air traffic control simulations, such as the simulator at Ottawa International Airport in Canada are used to train Air Traffic Controllers. They are also used in the design of new runways and in determining the designation of routes for the aircraft. To create a realistic training simulation, instructors act as recipients for messages and pilot the aircraft in the simulation; there is usually at least one instructor for each trainee. Therefore, the cost of training these air traffic controllers is high, a price that less essential services might find unacceptable. A presentation and demonstration of the simulator at Ottawa International Airport was arranged as a part of the Summer Computer Simulation Conference, SCSC '95.

Findler (1991) has researched the use of a Multi-Agent-System (MAS) to assist the Air Traffic Controller. Each aircraft is to have an agent that computes a solution to any approaching traffic, each agent's solution is fed back to a central controlling computer which decides upon the best solution for all the aircraft, while complying with the normal, turn to the right, regulations. The distributed nature of a MAS allows a large amount of processing to be performed by the remote agents, keeping the central work to a minimum.

2.3.2 Missile simulation

Missiles are expensive and are used once, therefore, simulation is a valuable tool. The trajectories the missiles will follow and the effects of design changes are simulated before the actual missile is tested. In order to obtain high performance and analogue environmental changes, hardware in the loop simulations have been widely used in this work (Curry and Combs 1995).

2.3.3 Car simulation

The Transport Research Laboratory (TRL) is engaged in research into many forms of road safety. One area of research is to determine the effects of alcohol. Volunteers perform a number of tests, the results of which are logged onto a floppy disk for analysis. One of the tests observes the responses of the subject when following another car in a simulator; computer controlled cars pull out from junctions, slow down, brake and speed up. Earlier work on this used a simulator with a continuous straight road and some work was performed in conjunction with Leeds University. A new simulator manufactured by Maritime Dynamics, based upon Silicon graphics Crimson Reality Engines has now replaced this and simulates the actual TRL test track.

Agent architectures were explored by Fergusson (1994) using a car simulation. The cars had to give way at traffic lights and manoeuvre around bends. This work is described briefly in chapter 3.

2.3.4 Marine Simulation

2.3.4.1 Port Simulation

Shipping companies obtain ever smaller profit margins; the halcyon days of the oil boom in the seventies are long gone. In order to compete, ports have to offer highly competitive rates. As can be seen around the Severn estuary in the UK, small changes, such as a rise in the Severn Bridge tolls, can make ports such as Cardiff, Newport and Barry less attractive than Bristol and Avonmouth; companies such as Geest and Bell line have moved their operations away from South Wales. Cargo damage, down time and the rates of cargo loading and discharge are particularly keenly observed by the ships' operators. Simulation can help to: minimise dredging costs; determine the optimum mooring configurations for ships on a regular trade; select efficient port designs and modifications; minimise environmental and traffic damage; minimise risk of collisions; maximise traffic throughput; discover cargo bottlenecks and streamline cargo operations; Therefore, both the ports' operators and the ships' operators can, more often than not, quickly recover the costs of simulation. In the UK companies such as Hydraulics Research, in conjunction with Maritime Dynamics offer port design consultancy; the second Severn crossing was assessed by them, both to reduce the hazard of accident damage to the bridge by shipping and to find an effective navigation aid for ships passing between the bridges.

Misuse of cargo handling equipment results in the wires, brakes and motors of the equipment having to be replaced more frequently. The delays while the repairs are carried out reduce the overall throughput of the port and increase the lay time for the ships. The goods being stowed/discharged will also be damaged leading to insurance claims. Crane cab, fork lift truck and Caterpillar tractor simulators (Caterpillar 1995) are being used to test new vehicle designs and to train the stevedores so that cargo is loaded faster and with less damage. A generic crane simulator has been built by Maritime Dynamics Ltd. This should result in faster turn around times for the ships and less claims for cargo damage.

- Discrete event Port simulations

Discrete event simulations are particularly suited to modelling the flow of cargo through a port. Simulations have assessed such diverse problems as estimating the time to embark an army through a port (Nevins et al. 1995) to determining where to position new terminals.

2.3.4.2 Model based simulation

The relative costs and effects of breakwaters, mooring arrangements and dredging can be considered. Water tanks are used at MIT and many other research centres for testing ship design changes. MIT has recently been experimenting with a submarine that swims like a porpoise, which is said to reduce friction and to be nearly silent. Following the spate of RO-RO ferries that have capsized, tank tests are also being used to evaluate car deck subdivision and sponsons[2] as a means of maintaining stability. In addition to ship manoeuvring, environmental effects of sedimentation and coastal erosion are considered using wave generation in water tanks at sites such as Plymouth University. The Port Revel (Guicharrousse 1990) simulation centre in France is a leader in the use of manned models of ships for ship handling training. However,

[2] Sponsons are buoyancy chambers fitted to the side of a ship to increase buoyancy in the case of a severe list developing. They are particularly associated with the dangers of hull integrity being breached on Roll-On/Roll-Off passenger ferries.

this type of training facility is rare, only three now exist, since the recent closure of a similar training facility at Little Creek, Virginia belonging to the U.S. Navy (Meurn 1996).

A recent study at the Port and Coastal Research Centre of CEDEX in Madrid (Iribarren et al. 1995), considered the port of Cadiz. Changes to the depth of the approach channel and the port layout had created a severe response to waves with long oscillation periods. A 1:125 scale model of Cadiz in a number of configurations was tested to determine why the changes had produced the response and then short and long term solutions were tested to improve the situation. The use of simulation before the earlier changes could have saved excessive down time for the ships operating in the port and prevented considerable damage to the port structure itself.

2.3.4.3 Helmsman Training

The Maritime Dynamics helmsman trainer, used at the HMS Raleigh shore base, emulates the steering stand of a warship. There is a mathematical model of the ship's performance characteristics, an interface consisting of a ship's wheel and physical models of generic instruments found aboard a real ship. These instruments, such as the compass, helm indicator and rudder indicator, are computer driven via digital to analogue hardware converters.

2.3.4.4 Desktop Trainers

Simpler than full mission simulators, desktop trainers can run on stand alone PC's or be networked to an instructor station. These systems are often used to supplement training on a full mission simulator. The simplest of these systems acts as a straightforward Computer Based Training (CBT) tool. For example, the Transit Satnav Receiver, developed by the author, allows the student to practise setting up and retrieving information from an electronic navigational aid; specific dangers in using the equipment are simulated and the satellite frequency can be set so that the training can be performed faster than when using real equipment. 'CAPTAINS', developed at Maritime Dynamics, is an example of a more interactive system. A hardware control box allows the student to set the ship's telegraphs and helm in order to navigate a ship, avoiding other vessels and land hazards. A simple radar display, offering head up and compass stabilised views, is used to plot the ship's position and track the targets. At the next level of complexity, a desktop ARPA simulator consisting of networked Archimedes computers is in use at Freemantle, in Western Australia. The displays are more realistic and the instructor can monitor the exercise. The cost is remarkably low and terminals can be added as required, to expand the system. Such systems are now being installed in China (Wang and Li 1996) for Global Positioning System (GPS), engine room, radio communications and ARPA simulations.

Liquid cargo operations trainers have been developed by Ferranti; LICOS (Ferranti 1985) is able to simulate operations for a number of cargoes including Liquid Petroleum Gas (LPG), Crude Oil, Petroleum Products and Chemicals.

2.3.4.5 Marine Bridge Simulators

A bridge simulator normally consists of at least one simulated ship's bridge and an instructor station. The bridge will consist of a number of components found on a real ship's bridge. Additionally, there will be a number of screens displaying the visual scene around the ship. A few of the more expensive systems may be mounted on motion platforms.

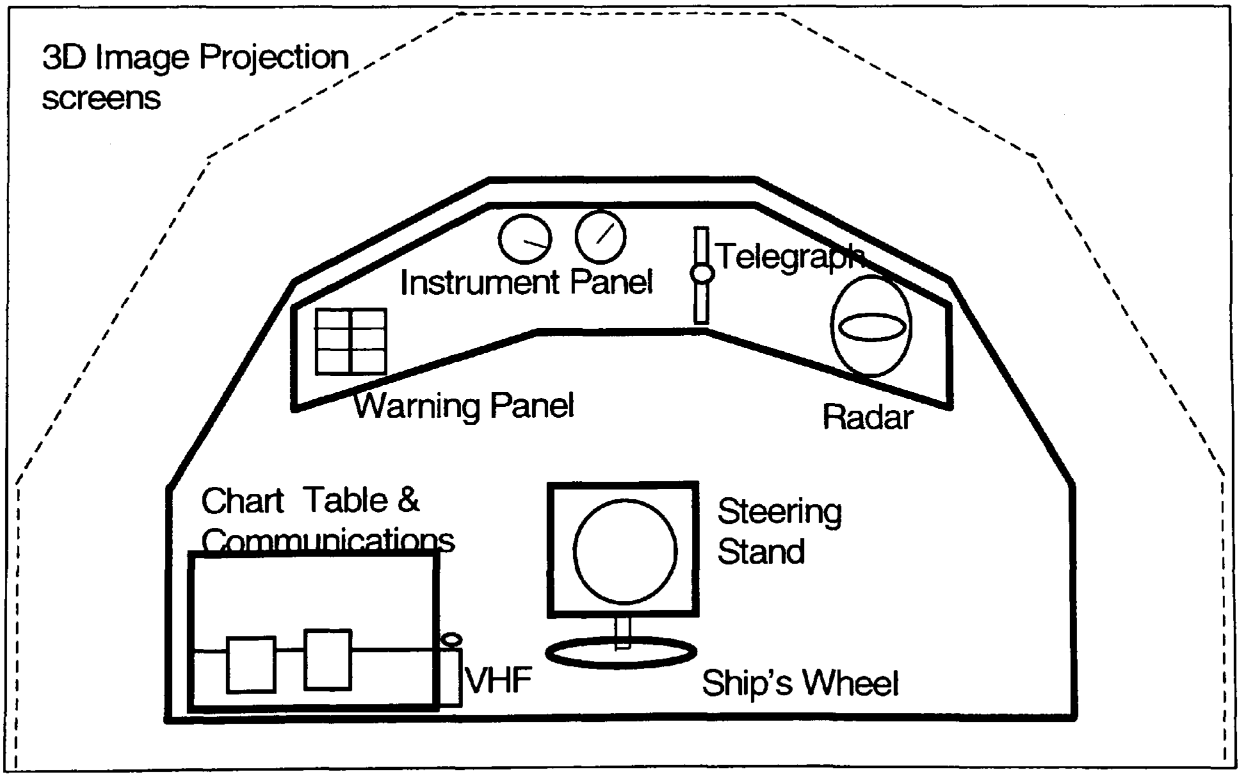


Figure 2-1 A diagrammatic view of a typical own ship bridge

- Visual displays

The creation of a reasonably flicker free animated display, for marine systems, requires about twenty five frames of visual information a second. On a marine simulator it is seldom possible to pre-compute scenes or use photographs of the foreground scene. One exception is the use of bow images in marine simulators. Photographs and bitmaps are also used to texture map scenes and provide distant backgrounds. In any case, scene generation is computationally expensive and used to be the domain of high end systems, such as, Silicon Graphics Crimson Reality Engines. In the early 1990's a few manufacturers used to supply less costly hardware to simulator developers, for example, Miriad Solutions I860 cards, Texas Instruments Tiga II cards etc.. However, the onus was mainly on the developer to create suitable programs.

Inexpensive graphics cards, costing under £200, are now becoming available with limited 3D hardware functions for rendering in real time. Examples include, the Matrox Mystique and Diamond stealth 3D 2000. Most of these cards will support standard interfaces, such as Microsoft's Direct3D API. The low cost will no doubt lead to a new breed of desktop graphics machines and cheaper simulators in the near future.

- Bridge equipment

The equipment used in bridge simulators consists of a mixture of types: real equipment that is stimulated by a computer; physical facsimiles of the real equipment; and displays that simulate the real equipment. Most simulators contain a mixture of the three types of equipment. The proportions of each style of equipment vary according to the budget and use of the equipment. Real equipment is used for items that are relatively inexpensive, more frequently used or more difficult to simulate.

For example, in the MARDYN fishing simulator at Kaohsiung Nantzu Maritime College the track plotter, the echo sounder and the side scan sonar are genuine equipment, that might be found on a fishing vessel. The ARPA[3] simulation (Moon 1991) is a computer simulation of a generic ARPA(IMO-SOLAS 1980), somewhat similar to real equipment that is produced by Furuno, Kelvin Hughes and Racal Decca,

[3] In a marine context ARPA is an acronym for Automatic Radar Plotting Aid. In most modern ARPAs the plotting functions form an integral part of the radar equipment, the complete radar now being called an ARPA.

well respected manufacturers and suppliers of real ARPA equipment, widely used on merchant ships and fishing vessels. The keypad for the generic ARPA is a physical facsimile of the keypad used on real equipment. This is a cost effective mix of equipment that has a realistic look and feel.

- Submarine bridge simulation

Marconi demonstrated a submarine simulation at the Computer Graphics Show, London, 1992. The full simulator is highly realistic using real equipment and a single computer generated visual channel via a periscope attached to a Silicon Graphics machine. A wide variety of sonar simulation for submarine detection is performed by Thomson Marconi Ltd. (Thomson Marconi 1996). Previously known as Ferranti Computer Systems (Ferranti 1985) , the company supplies trainers that are used both ashore and on-board ships.

- Surface ship bridge simulators

Large simulator installations such as **CA**rdiff **S**hip **SIM**ulator (CASSIM) were costly. CASSIM was based upon PDP-11 computers and was a very advanced system when it was first installed in the late 70's. This genre of simulator is described by Caillou (1991) using the NORcontrol simulator at St. Malo as an example. The Canadian Coast Guard College simulator, developed by NORcontrol, at Sydney, Nova Scotia is one of the latest of these large simulation systems.

As computing power has increased, and with the closure of many of the larger marine training colleges, less costly simulators based upon networks of desktop and workstation computers have become popular. The Microsim radar simulator at Fraserburgh in Scotland and Maritime Dynamics Port Design and Fishing simulators at Kaohsiung, Taiwan are examples of this type of simulator. Transas Marine now offer a software solution that will run on PC-compatible computers. The hardware configuration can be designed and fitted by the simulation centre, allowing a gradual, modular build, as funds become available. The IDESS International training centre at Subic Bay, in the Philippines have developed their simulator in this way. The system has the added advantage of coupling to their ECDIS chart display and Furuno radar display. These are electronic navigational aids as used in commercial ships, they give the simulator a realistic interface at a reasonable cost. The latest option is an oil pollution simulation that estimates the spread of an oil slick; this can help in determining the most effective method of cleaning the slick. The resulting slick is visible on the simulator visual screens.

2.3.4.6 The Simulator Instructor

It is important to convince the students in the simulator that the exercise is close to the 'real thing'. Possibly the most important aspect of this is the instructor's capability. The exercises must be designed to use the strengths of the simulation and hide the weaknesses. Firstly, a realistic complement of crew must man the simulated ship's bridge; if too many people are permitted on the bridge then the rigid team structure, found on a real ship, breaks down and anarchy prevails. Secondly, the exercises must tax the crew's capabilities to the correct level; too simple and they will loose interest, too hard and they will make mistakes that may highlight the deficiencies of the simulator. Typically, these deficiencies include target ships that navigate across land or through shallow water, etc. Once the students obtain an impression that the exercises are not realistic in this way, then they may lose confidence in the simulator. Thirdly, at run-time the instructor must be able to respond realistically to radio communications and perform target ship manoeuvres as necessary. This can be very demanding, particularly if several student ship bridges are attached to the simulator.

As mentioned in the introduction, the instructors of many new 'micro simulators' perform the task as a small part of their normal duties. Although these instructors are navigators and teachers, some with many years of experience, they may have little experience of using computer simulators. Chen (1992) found that "*many of these instructors felt unable even to operate the instructor station*".

2.3.4.7 Instructor Stations

The exercises that are performed on the simulator are set up and monitored by an instructor. In the larger simulators the instructor does this from a terminal in a separate room. Standard features of these instructor stations allow the instructor to: select a simulation area; position computer generated target ships; set the courses and speeds these ships will follow and set the environmental conditions such as the wind speed and range of visibility.

The instructor terminal interfaces vary widely. Some consist of displays showing the numerical states of the ships and other salient information, possibly supplemented by repeaters showing the same view as the student bridge. While this type of display is not immediately intuitive, experienced instructors are able to alter system parameters quickly and the text updates rapidly, taking little computing power. Another interface is a 2D plan display of the scene, where the instructor has an unrestricted view of the situation, not limited by the simulation visibility. It is easy for the instructor to understand such a map. However, this global view gives the instructor the opportunity to make decisions that would have been impossible from the bridge of the ships being controlled. Typically, the instructor may avoid a ship before it could have been detected by an actual navigator, or more likely, take into account future dangers, invisible to the navigator when assessing an avoidance manoeuvre.

On many simulators the instructor pre-plans the routes of the target ships in order to ease the need for continuous target monitoring and manoeuvring at run-time. This is effective for short exercises where the position of the student ship can be predicted reasonably accurately. However, in longer exercises students may navigate without encountering the instructor's planned situations, for example, by taking a slightly longer track or manoeuvring at less than optimum speed. Another problem with this type of target control is that the ships often follow the pre-planned routes without altering their headings for the set and drift of the current or deviating for other ships, unless the instructor intervenes. Pre-planned exercises do, however, allow the same exercises to be repeated with several groups of students and the results to be compared.

2.4 Marine traffic management and collision avoidance

2.4.1 Traffic management by shore stations

In busy coastal areas, such as the Dover straights, traffic separation schemes are used to reduce the danger of collision. A large volume of traffic will be passing through the busy area, en route to other ports. This traffic is routed so that vessels travelling in opposite directions do not meet; the routes may be likened to the lanes of a motor way on land. Often, these schemes are monitored from the shore, by radar. A local authority, for example, the coast guard, watches for ships that contravene the regulations. A computer analysis of such a marine traffic system has been made by Redfern and Lin (1995).

These traffic schemes usually include some form of reporting system. The ships log in on entry to the scheme, may be handed from one authority to another as they progress through the scheme and finally log out when they depart from the scheme. There are several advantages to this: any ship that fails to log in is identified as a potential problem before it reaches the busy areas, a helicopter can be dispatched to discover the ship's identity if all else fails; the last known position of a ship is updated so that if a ship is overdue, the search area can be narrowed down; approaching craft can be made aware of large or unusually hazardous vessels that are navigating in the scheme; and, if a collision does occur, the size of the ships, the types of cargo, the number of crew members, etc. are already known, thus speeding up the emergency services response.

2.4.2 Collision Avoidance On Board Ship

On a ship collision avoidance is performed by the navigator. The International Regulations for the Prevention of Collisions at Sea (IRPCS 1989) have evolved over many years and are used by the officers of ships of all nations. These rules are very effective in resolving potentially hazardous situations involving only two ships in unrestricted waters. However, local rules may apply in certain areas, particularly rivers and other inland waterways to supplement or replace the IRPCS (1989). In clear weather and daylight the compass bearing, aspect, estimated range and the shapes carried by an approaching vessel are used to determine the risk of collision. In addition, on a clear night, the lights a vessel displays assist the navigator. Radar can also be used to augment the navigator's assessment; trigonometry being used to determine the level of danger that the target ships present. Modern Automatic Radar Plotting Aids (ARPAs) perform the calculations and display the target ship's course and speed vector, or other similar indications of the danger to the ship. For example, some of the Sperry systems display Predicted Areas of Danger (PADS); circles or ovals on the radar screen which represent areas that your ship should avoid if all the ships maintain their present courses and speeds.

2.4.3 Domains

The concept of domains (Goodwin 1975) has been a major factor in the development of automatic collision avoidance systems based upon computers, a domain being *"the effective area around the ship the navigator would like to keep free from other ships and stationary objects"*. Prior to Goodwin's work there was no formal way of determining the necessary separations between ships; Goodwin observes that *"nothing has been suggested for separations between ships but it is likely that ship domains do exist, maintained voluntarily and almost unconsciously by navigators"* (1975). Goodwin suggests a number of factors that decide what the domain would be. The three main headings were: psychological factors, such as the navigator's experience; physical factors specific to one ship, such as size and manoeuvrability; and physical factors general to all ships, such as weather and traffic density.

Goodwin studied the paths of ships in the area around the 'Sunk' light vessel to show that these domains did exist. This study gave the statistical average size of domains, based upon observing ships in the area around the light vessel. However, the shape of the domain was affected by the relative position of the other traffic. The navigators wishing to keep a larger area of free space on the starboard (give way) side of their vessel and needing less free distance astern. This gave a discontinuous shape to the domain. Knowledge of the size and shape of these domains used by real navigators has been valuable to the designers of expert systems. However, the problem of modelling a domain that is not a smooth shape needed to be addressed. Since the domain represents the area a navigator tries to keep empty around the vessel, to do this the navigator must plan the future long before an approaching ship enters the domain, and this made further research necessary.

2.4.4 Arenas

Goodwin's work was supplemented by Davis (Davis et al., 1980), who introduced the concept of arenas. The arena is an area where the navigator will begin to consider action to keep the domain clear. Using statistics derived from a questionnaire, Davis showed this to be approximately twice the size of the domain. The actual figures suggested a decision distance of 4.3 nautical miles on the starboard side and 2.6

nautical miles on the port side. This gave an arena radius of 2.7 nautical miles with the ship offset 1.7 nautical miles 199 degrees from the centre, as shown in Figure 2-2.

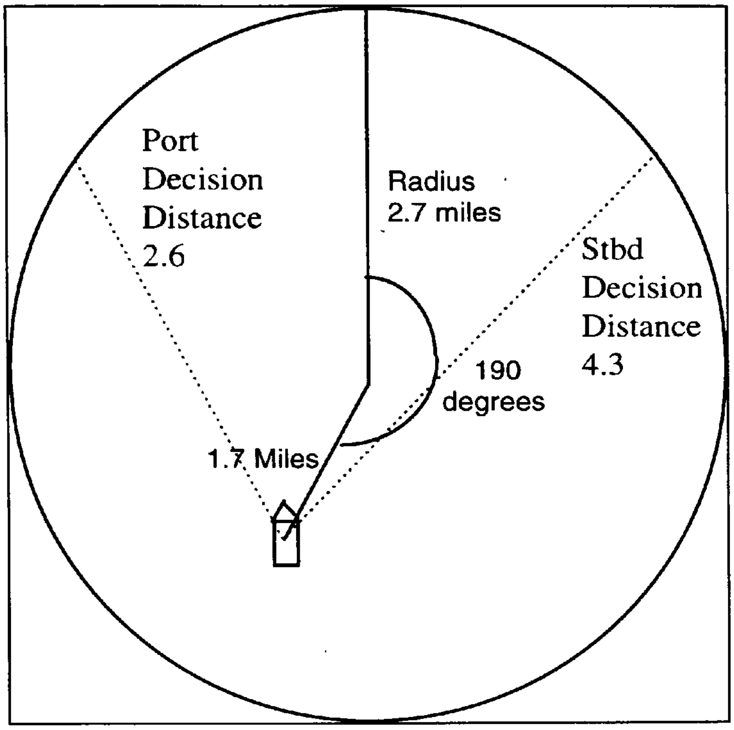


Figure 2-2 Arena as suggested by Davis et al.

2.4.5 Risk Heuristics

One of the problems with the concept of domains and arenas is that they take statistical samples and produce average results. The results are valid for an average type of vessel, manoeuvring in an average manner, in average environmental conditions; they are based upon *"uniform 'ambient' distribution"* (James 1986). Unfortunately, few ships will fit this average profile. Domains and Arenas provide a useful starting point for research into collision avoidance, however, a navigator will actually apply much more refined criteria. Considerations such as the relative speed of the ships, other nearby hazards and the type of hazard encountered will play a part in the deliberation. The relative speeds of the ships were taken into account by Colley et al. (1983) improving the concept of arenas with the Range-to-Domain/Range-Rate time factor.

Following this, Smeaton and Coenen (1990) have developed heuristics for the problem of collision avoidance. These heuristics embrace the idea of a domain, although the domain is circular and changes dynamically according to a number of factors. The radius of this dynamic domain is called a 'domenge' and a heuristic formula is given to calculate the value of the 'domenge' for any encounter. In a two ship encounter, if a vessel is going to enter the circle subscribed by the 'domenge' then risk is said to exist. An Arena is created as a multiple of the domenge, thus this Arena will also vary in size according to dynamic factors. A heuristic formula is given to determine a list of targets ranked according to danger, use of this formula has been made within the MARINES project.

2.4.6 Intelligent guidance systems

Burns (1995) has considered the possibility of fully automatic intelligent integrated ship guidance as a means of providing a *"cheap cost effective, efficient service to the customer"*. This paper briefly presents some methods that may offer a complete method of automatic navigation. Some interesting points are raised about collision avoidance, such as, the use of Manoeuvring Regimes (MR), Tactical Regimes(TR) and Strategic Regimes (SR). A MR is similar to a domain and the SR similar to an Arena, however, the TR is newly identified. A TR describes the distance at which the navigator will actually make the alteration of course. The concept of the TR seems reasonable and provides a sensible hypothesis for further research. Should the TR exist, then navigators will maintain them almost unconsciously. As with domains and arenas, some form of statistical study will be needed to show whether they are indeed *"concentric circles"* (Burns 1995). If this is so, it is probable that different weightings, for different approach sectors (Goodwin 1975) will apply.

2.4.7 Automatic collision avoidance using a computer

Automatic collision avoidance requires a completely prescriptive method of determining an avoidance action or the inherent ability to adapt to changing circumstances. The International Regulations for Preventing Collision at Sea (IRPCS 1989) do not provide a complete rule set that can be directly considered using a computer. James (1986) mentions some of the qualitative problems involved.

2.4.7.1 Using fuzzy logic

The inexactness of the collision regulations lends itself to the idea of fuzzy logic; *"it is possible to view these regulations as defining a set of fuzzy goals and constraints"* (James 1986). James goes on to describe the use of such sets for avoiding fixed structures, and shows that the results are consistent with the concept of domains. However, while the discussion of fuzzy logic for ship interactions does provide statistically sound results, there is no examination of the compliance with the regulations. The results are only for a small selection of manoeuvring situations and the requirements for safe avoidance have been relaxed.

Lisowski (1992) develops fuzzy probability sets based upon approach sectors chosen empirically and statistical information from navigators. However, the results of using the sets are not described.

2.4.7.2 On board real ships

Grabowski (1990) at Rensselaer Polytechnic has developed an expert system for large vessels navigating in the environs of New York Harbour. The system provides decision support for the officers of the watch.

Following the EXXON Valdez stranding, Grabowski has been working with EXXON tankers on a similar, although more sophisticated system (Grabowski and Sanborn 1992), to improve safety for vessels in the Gulf of Alaska. Special conditions can apply in this area with ice and snow making accurate radar fixes difficult to obtain. The Shipboard Piloting Expert System (SPES) forms an intelligent part of a complete integrated ship bridge system, Sperry Marine's ExxBridge.

The latest research is for a distributed piloting system for the St Lawrence Seaway (Grabowski and Sanborn 1995b; Sudhendar and Grabowski 1996).

2.4.7.3 Using discrete event simulation

Smeaton and Tucker at Liverpool John Moores University have used a collision avoidance system (Smeaton and Coenen 1990) as an advisor for the students in a bridge simulator. Grabowski and Sanborn (1995b) have also performed tests on a simulator using the SPES system.

2.4.7.4 Using continuous simulations

An intelligent interactive environment for a maritime real-time expert system has been created by Blackwell's group at Plymouth University, simulation is used to assess the performance of their collision avoidance system (Blackwell et al. 1988). The system developed is intended for use as an expert advisor aboard a real vessel, in much the same way as the work of Grabowski's and Smeaton's groups. The original implementation of this system was on a single Atari ST computer.

Blackwell's system is perhaps the closest in concept to MARINES; the final version (Blackwell et al 1991) running on a multi-tasking Acorn Archimedes RISC computer.

There are several similarities between this system and MARINES:

- Several processes are connected using a message passing architecture and run pseudo-concurrently.
- The processes provide 'intelligent' collision avoidance through the use of an inference engine and production rules.
- The system can use different rule bases.

However, there are also several fundamental differences:

- The simulation performs collision avoidance, however, no track-keeping capability is provided. Therefore, the effect of set and drift of the current is ignored.
- The system is focused around the 'own' ship[4]; the display is specifically designed to follow the own ship. The system acts as a test bed for validating the manoeuvres of the own ship in accordance with the collision regulations and safe practice, thereby, checking the expert system.
- The 'hazard'[5] ships are used to create situations to test the system. The accurate motion of these ships is not of primary consideration. Simple parameters are used like position course and speed.
- The 'hazard' ships only try to manoeuvre for the 'own' ship, they do not try to avoid each other. Also, When the hazard ships are too far away (if

[4] The 'own' ship is considered to be the only fully functional, automatically controlled ship in the simulation. This represents a real ship fitted with the Expert Avoidance System.

[5] Blackwell differentiates between 'own' ship and 'hazard' ships. The 'hazard' ships provide the dangers that the own ship has to avoid. Each 'hazard' ship uses a simplified expert system and only avoids the 'own' ship. Hazard ships are somewhat similar to the computer generated target ships used in MARINES. The 'own' ship in MARINES would be piloted by the students under instruction.

the Range-to-Domain/Range-Rate(RDRR)[6] is too great) they are not considered. Therefore, actions will not change the future events in the simulation.

- The simulation steps forward in coarse 20 second time increments and is synchronised by regular messages.

Blackwell's system is an ideal test bed. It allows: the configuration of parameters; repeatability of experiments; many experiments to be conducted in a short space of time. However, it reduces the realism and complexity of the simulated world in order to achieve this.

MARINES, on the other hand, permits some complex interactions to take place. The ships are mathematically modelled and are subject to natural conditions, for example, the auto pilot settings alter the manoeuvring and yaw characteristics. Track keeping is performed by the agent which adjusts the ships course at run-time and this reduces the accuracy of target ship information; which is based upon interpolation of historical data and relies upon a steady course and speed. The situation will be altered by the set and drift of the current, the agent applying corrections to the ship's desired course. The time steps are not synchronised; each agent operates asynchronously, therefore, the information may change between their individual assessments of the situation. These factors prevent precise replication of exercises and make quantitative evaluation more difficult. However, this imprecise, realistic simulation raises some interesting issues, a real navigator would have to address, that may be suppressed in a straight forward test bed.

PC Maritime

The PC Maritime 'Officer of the Watch' simulator provides automatic collision avoidance for the target ships, based upon domains (Goodwin 1975). The simulator extends the work at Plymouth University (Blackwell et al.1988; Blackwell and Stockel 1989, 1990; Blackwell, Rangachari and Stockel 1991), although, only a very brief description has been uncovered about the system. It is unfortunate, given the advanced

[6] The Range-to-Domain/Range-Rate(RDRR) performs a similar function to the Danger Coefficient (Smeaton and Coenen 1990).

nature of the simulation, that more has not been published. From the limited description available there are several similarities with MARINES, however, the system is intended to replace the instructor in a relatively simple environment. One of the expert users interviewed has evaluated and used the system. The user found it to be a valuable and useable simulator but observed that it did not model current. According to PC Maritime themselves, a single rule base is used to control all the target ships and vessels can be flagged as rogue vessels, ignoring the rules.

2.5 Automatic track keeping

If the problem of navigating a ship through a narrow channel is simulated, then it is sometimes valuable for automatic track keeping to be performed. A physical description of the channel is provided and tidal depth variations and current set and drift may be applied. The ship model is usually provided by a mathematical force model (McCallum 1980). It is then the task of the analyst to devise a computer program that will automatically navigate the ship through the channel under varying conditions.

The primary use of this type of simulation is in port design. A large number of scenarios can be undertaken automatically by the computer and the results analysed. These can first be performed using a well tested model of a real ship manoeuvring in the present channel configuration. This permits the system to be tuned, then the simulated channel configuration can be adjusted for dredging or a new ship model can be tried out. The outputs from the computer pilot must also be compared with some results obtained by expert pilots navigating the same model in the same simulated channel.

Once tuned, an automated track keeper, such as this, may permit experiments to be performed at rates well above real time. Thus, a large number of experiments in a wider variety of trial conditions may be achievable far more rapidly and more cost effectively.

2.5.1 Mathematical

A track keeper has been developed at Maritime Dynamics based upon trigonometric functions. The mathematical ship model that it controls is quite sophisticated, taking into account shallow water effects, bank effects, etc. The calculation of helm over position is determined from the distance to the next way point, the speed and manoeuvring capability.

This type of approach has also been taken in MARINES, although the calculations have been developed independently and it is not known if the same equations have been applied. In MARINES only a simple ship model is used, and the track-keeping is designed for open water conditions. Another significant difference is that in MARINES the agent controls the auto pilot setting, rather than the ship's helm directly. This reduces the agents' communication with the environment but, when using a Proportional plus Integral plus Derivative (PID) auto pilot, has implications for the accuracy of the track keeping. The conventional PID auto pilot provides a reasonably accurate, reliable automatic controller for maintaining a pre-set course, it is not so good at making smooth alterations onto new courses.

2.5.2 Fuzzy logic auto pilots

In order to improve the course keeping characteristics of a PID auto pilot, when steering a straight course, an integral of the off-course angle is fed back to the controller to gradually bring the ship to the correct course. Tuned properly, this will make the auto pilot apply adjustments to allow for wind, transverse thrust, etc. It might, for example, apply a few degrees of permanent starboard helm to maintain course, if the ship is affected by a side wind consistently turning it to port. However, the integral will grow very rapidly if the ships heading is far from the desired course, causing large helm angles to be applied. This is undesirable, therefore a limit is usually applied to prevent the off course error signal gain from becoming too large. On some older systems, where this was not the case, the integral controller had to be disabled manually during large course changes. Many navigators will have observed the phenomenon, where having successfully settled the ship on a new desired course the auto pilot apparently developed a mind of its own. If the integrator was not switched off during a course change then the error signal became very large, trying to return the ship to course. In fact, the integrator on some systems operated in segments and might even apply the corrections in the wrong direction. If another segment was crossed then the error signal could rapidly swing the other way, which could cause the auto pilot to steer a completely different course and also apply huge helm settings to do so. This was found to be so dangerous that often the controller was left switched off; the officer of the watch performing the corrections manually. Furthermore, the PID auto-pilot needs to be adjusted for different ship types as well as load, trim and weather conditions. For this reason, PID auto-pilots seldom operate at their full potential.

Several studies have suggested the use of a fuzzy logic controller to replace the PID controller. Polkinghorne et al. (1995) showed a fuzzy logic auto pilot achieved a 90 degree turn 50% faster than a PID controller and had 25% less rudder activity. The creation of a self-organising auto-pilot based upon this fuzzy logic controller may provide a significant improvement in performance and also permit the same auto-pilot to be used in many different craft and conditions without manual readjustment. This fuzzy logic auto-pilot can also improve fuel consumption and reduce yaw.

2.6 The future of automatic marine collision avoidance and track-keeping

Zhao et al. (1992) reviewed the history of automatic collision avoidance and suggested future directions for research. Given the level of world-wide interest it seems odd that automatic marine collision avoidance has progressed relatively slowly.

One possible explanation is that, on its own, automatic collision avoidance has little value to ship owners. No doubt, most would welcome a fully automated bridge with a commensurate lowering of manning scales and costs. If the bridge has to be manned then an expert system adds cost and complexity with few visible benefits.

At the moment, piloting systems do not claim to equal or better a human's capabilities; navigators are qualified professionals and should be able to cope with any situation a computer can currently deal with. One area where the expert systems do offer some benefits is by providing support for the navigators decision. These expert systems contain the aggregated knowledge of a large number of specialists. However, as with any software, there may be errors, omissions and ambiguities in the system.

For many reasons it will be some time before an autonomous expert system takes a bridge watch on its own. These reasons include: cost; robustness and recovery from equipment failures; the need to meet the conflicting requirements of providing a completely prescriptive solution and complying with the current regulations; the potential damage that could be done by a system failure; and the need to cope with a wide variety of unplanned events. Furthermore, two of the most essential factors for general adoption of any marine navigation system are a legal requirement to carry the system in order to trade in a particular area, and a financial incentive to carry the system. The marine industry is quite conservative; radar was a huge step forward and was widely available in the 1960's, but it was not universally adopted until the 1980's. Legal requirements for equipment on specific ship types have gradually improved safety; most ships trading in the USA are required to carry radar, Loran, and GPS navigation aids. Better safety standards may also result in reduced insurance premiums. From time to time, the International Maritime Organisation (IMO) introduces international recommendations for essential improvements.

Grabowski's work has been supported by the U.S. Department of Transportation, Maritime Administration and Coast Guard as well as Exxon, Sperry and the National Oceanographic and Atmospheric Administration. The use of the system aboard real ships and, in particular, as a part of an integrated navigation system, presently makes it arguably the most advanced system. This commercial and administrative support is necessary if such systems are to become widespread.

Automatic track-keeping and improved auto pilots do have potential for industrial applications in the very near future. Intelligent track-keeping, combined with GPS systems has the potential to minimise the distances travelled, saving large sums of money and time for ships on world wide trades. If weather routing is also integrated then optimum routes with minimum damage to the ships and cargoes may be attainable.

If fully automatic ship navigation does become commonplace then it is possible that the traffic monitoring schemes will have to develop into ship traffic control centres. A distributed or Multi-Agent architecture may well be necessary to accommodate the high level of computer activity required for such a task. Grabowski and Sanborn (1995a) are researching distributed automated piloting in the St Lawrence Seaway and the use of MAS for air traffic control is being considered (Findler1991). There is potential for a combination producing a MAS for marine navigation.

Overall, it is the combination of traffic monitoring, track-keeping, automatic collision avoidance and improved auto-pilots into a single integrated system that is sought. Together with many other safety improvements, this has the potential to reduce marine accidents, cut costs and improve efficiency. In addition to the objectives for simulation and Multi-Agent Systems MARINES provides a platform that brings automatic track-keeping by auto pilot and automatic collision avoidance together. New agents can be created that use different techniques and rules for comparison.

2.7 Characterising the MARINES system

The MARINES system is a continuous simulation of a marine simulator instructor station. However, as Wildberger (1995) suggests the boundaries between discrete event and continuous simulation are blurring. The use of an expert system and object based event-driven programming techniques permits a mix of continuous and query driven simulation. Asynchronous timer initiated events start the agent deliberation, the agent will then plan pseudo-continuously and autonomously until it has completed the current plan. The ships, on the other hand, are always updated pseudo-continuously. Since Windows 3.1 is a co-operative Multi-Tasking environment and is not Multi-threaded, a truly continuous simulation containing several processes is not possible. However, the update cycle and the move cycle are not synchronised and separated for all the processes; the overall effect is of a continuous simulation running at a rate proportional to real time.

MARINES provides limited automatic collision avoidance and track-keeping for computer generated target ships. Although the implementation is relatively simple, the collision avoidance in MARINES is developed using the risk heuristic approach. This approach was described by Smeaton and Coenen (1990) and referred to by Grabowski and Sanborn (1992), two of the most advanced avoidance systems.

The track-keeping algorithms in MARINES rely upon straight forward algebraic solutions. This is conceptually similar to the track keeper developed at Maritime Dynamics for Hong Kong University and differs from the work of Kasasbeh (1994) and Burns (1995) where fuzzy logic is applied.

2.8 Summary

Simulation can be broadly divided into the fields of continuous simulation and discrete event simulation; continuous simulation is often used for real time systems such as aircraft cockpit trainers, while discrete event systems may be used to model throughput of a system in fast time. The time line of a discrete event simulator is not necessarily proportional to the time that has elapsed in the real world, whereas the time line of a continuous simulation usually is proportional to real world time.

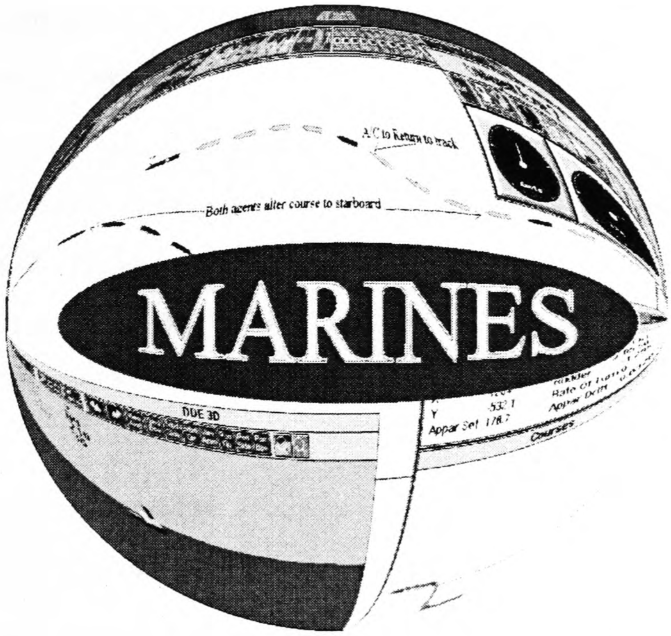
Many aspects of marine simulators are common to a number of other training simulators. A few of these similarities have been highlighted in this chapter, for example: the visual systems in aircraft simulators are in some cases the same as those used in marine systems; some Desktop trainers include a separate instructor station.

Air traffic control and car simulations have been used to explore the use of Multi-Agent Systems' technology. Conversely, it is perceived that MAS technology may be beneficial to marine simulation.

The instructor is possibly the most important factor in creating a believable marine simulation. Well designed exercises, a realistic bridge team and the ability to respond rapidly to a wide range of student manoeuvres and communications make the best of an imperfect simulator. An instructor station that provides some automatic assistance for novice instructors is desirable.

Marine traffic management and collision avoidance have been the subjects of many studies. Traffic separation zones have improved safety in many busy coastal areas. Unfortunately, the automation of collision avoidance has been more elusive, although concepts such as domains and arenas have been instrumental in understanding the behaviour of navigators and prototype expert systems have been developed to assist bridge teams.

Chapter Three



Multi-Agent Systems

3. Multi-Agent Systems

3.1 Overview

The terms 'agent' and 'agency' are now widely used in many areas of computing. Unfortunately, this has led to many varied and incomplete, if not conflicting, definitions of what an agent is. As far as the author knows, there is no globally accepted, inter-disciplinary, definition of the term agent that is sufficiently detailed and specific enough to be meaningful as a description of a computing methodology. This chapter provides 'bounded' definitions of 'agents' and 'agency' that apply to the MARINES research.

Specifically, the agents described in this thesis form part of a Multi-Agent System (MAS), a sub-field of Distributed Artificial Intelligence (DAI). A discussion of past and current MAS theory and practice is intended to highlight some of the features that researchers have found to be necessary properties of an agent in a MAS. The relevant features are then combined to give an indication of what gives a MAS agent 'agency'.

Following this, the system architectures used in building complete MASs are described and then several different internal agent architectures are discussed.

There is then a section on tool kits that are available to assist in creating MASs. This is followed by a section describing some previous simulations that have been created using MAS. In general, these have been used to construct the changing worlds that agents inhabit in order to test aspects of agent behaviour. Even relatively sophisticated simulations, such as Phoenix (Cohen et al., 1989), are not really aimed at solving a real world problem of fire fighting, rather they are constructed to give an experimenter the ability to test an agent under controlled conditions while solving a non-trivial problem.

3.2 What is a Multi-Agent System (MAS) ?

A MAS has been described as *"a loosely coupled network of problem solvers that work together to solve problems that are beyond their individual capabilities"* (Durfee et al., 1989). This section tries to give a history of how such systems have developed, and describes the features and characteristics exhibited by agents within a MAS.

3.2.1 A brief history of Multi-Agent Systems

There follows a short history of agents and a description of agents used in Multi-Agent Systems, which, it is hoped, will give a reasonable foundation for understanding the agents used in MARINES. Most of the terms are explained in more detail in the body of the text.

Minsky (1986) supplied one of the most widely referenced descriptions of computer agents. These agents *"produce an effect"* and may exhibit some intelligence; many simple agents working together may produce results that are greater than the sum of the individual agent's capabilities. The derivation of MAS research from A.I. and expert systems research has led to the use of other A.I. techniques in the implementation of MASs. Blackboard systems (Maitre and Laasri 1990), production rules and neural networks (Adams and Nabi 1989; Alpaydin 1993) have all been used. Some of the agents have been more complex and a distinction has been drawn between coarse grained systems, with fewer, more complex agents, and fine grained systems, with more, simpler agents (Maitre and Laasri 1990).

Some major areas of research in MAS have concerned the communication between the agents (Okada et al., 1993), co-operation between agents (Wooldridge and Jennings 1994), planning(Von Martial 1990), agent architecture(Fergusson 1994) and agent frameworks (Hanks et al., 1993). Several agent languages have been created and the different types of messages described in speech act theory(Searle 1969) have been used as the basis for some of this research. Agent Orientation (Shoham 1991, 1993) describes one such language and how agents employ Object Orientation. Agents and objects have more than a passing similarity, since communication and the encapsulation of data and functionality into a single entity are aspects of both. Agents usually display at least one further attribute, that of autonomy; once set into action most agents are able to perform tasks without continuous user prompting.

Changeable environments such as TileWorld(Pollack and Ringuette 1990), TruckWorld (Hanks et al., 1993) and Phoenix(Cohen et al., 1989) have been used to assess MAS agents and these are described towards the end of this chapter.

3.2.2 What is a MAS agent ?

The term 'agent' is in day to day use in the English language and the meaning, on its own, may be interpreted in a number of subtly different but related ways. The dictionary definitions include *"performing a function on behalf of another"* and *"creating an effect"*(McCleod 1987). Artificial Intelligence *"aims to construct agents that exhibit aspects of intelligent behaviour"* (Wooldridge and Jennings 1994). From a Distributed Artificial Intelligence (DAI) perspective *"agents do things, they act"* (Wooldridge and Jennings 1994).

Such a varied definition has disadvantages when naming a computing methodology; most, if not all, computer programs can be seen as containing agents (Moffat and Frijida 1994). Is a compiler an agent ? Is a spell checker an agent ? The answer is probably yes, if you consider it in that way; each creates an effect and performs a function on behalf of another. However, Castelfranchi (1994) suggests that an agent should not only be a means of achieving something, but should also be goal directed and autonomous. It seems that the more the term agent is used the less clearly it is defined, *"In the AI literature there is not always a clear distinction between an agent and a process"* (Pebody 1993). Furthermore, the term agent is used in a variety of computing fields including DAI, Computer Supported Co-operative Work (CSCW), Robotics, Software Engineering and Information Systems. If all the programs claiming to include agents are considered, then there is no one property that sets them apart from other branches of computing. In order to define the terms 'agents' and 'agency' for the purposes of this thesis, the agents are classified according to a combination of the features found in an agent and the computing application that the agent is designed for. However, features cross inter-disciplinary boundaries, more and more frequently, as the quantity of agent research increases. Some of the attributes that have been suggested as contributing to agent-hood are outlined below. The attributes considered particularly relevant to this thesis are then listed at the end of this section.

Some agents have been ascribed anthropomorphic features and characteristics; a pictorial representation or a particular human trait, as in Letizia (Leiberman 1996). The term 'avatar' (Halfhill 1996) has been applied to these agents to distinguish them from DAI style agents. Information systems use *agents* to perform search and retrieval tasks from data bases. In general, these systems consist of a number of individual agents (Robinson 1991) lacking the communication and co-operation found in a DAI MAS. Interface agents and information search agents are also considered here as falling into a separate category from DAI agents; the foundations of the research for information search agents and interface agents have developed more from the anthropomorphic and intelligent attributes of agents than from the co-operative, collaborative approach of agents in MAS. It should be noted, however, that on the Internet, with the advent of languages such as Java, multiple communicating, co-operating, intelligent agents are expected to run as distributed processes on a number of machines.

To further help in distinguishing the types of agent certain adjectives have been used, for example, 'intelligent agents' or 'autonomous agents'. Unfortunately, each chosen adjective, on its own, may be applicable to a wide range of agent systems and the chosen adjectives themselves have led to further debate; what is an 'intelligent' or an 'autonomous' program ? The use of these terms is discussed further in the body of this chapter.

MARINES is designed as a Multi-Agent System (MAS) within the sub-field of DAI and it is this type of agent that is described here. It should be noted that the terms agent and agency have also been discussed at a number of MAS workshops without a complete consensus of opinion. Wooldridge has given a description of 'agents' and 'agency', as used in DAI (Wooldridge and Jennings 1994) and several conferences have held workshops on agent theories. Wooldridge also posed some questions that agent theories should address; *"What exactly are agents? what properties should they have...?"*. At the workshop where the question was posed these properties included *knowledge, beliefs, intelligence* (Wooldridge and Jennings 1994), *autonomy*, *cognition* (Castelfranchi 1994), *commitment*, *rationality* and *intentions* (Dongha 1994). Pebody (1993) also described such agents *"In DAI terms...an agent is an embodied entity that operates in the real world...an agent or process can be seen as a self contained unit* *with a set of inputs, a mechanism and a set of actuatory outputs"*. The conventional DAI view is that agents have *beliefs, desires and intentions (BDI)* and use these to plan co-operatively (Wooldridge and Jennings 1994). An alternative view is that of situated action (Wavish and Graham 1994), an agent will respond differently according to the situation that it is placed in. Taxonomies of Multi-Agent Systems have been suggested (Bird 1993; Van Dyke Parunak 1996) to subdivide MAS agents. These taxonomies describe some common agent features, such as communication and co-operation. Bird's taxonomy also gives an insight into some measure of agency, largely based upon co-operation. Additional agent properties can be derived, and further credence given to those already described by observing the major areas of research into MAS. Examples are: message passing (Craig 1991); planning (Von Martial 1990); and architecture (Fergusson 1994).

Thus, for the purposes of this thesis, a MAS agent is seen as having some of the following attributes: intelligence; autonomy; communication; co-operation; knowledge; adaptability; planning capability; reactive capability; beliefs; desires; intentions; encapsulation; sensors; and effectors. Not all of these qualities need to be present in every agent, although the levels of each property will provide some measure of the *agency* of an agent in a MAS. A MAS must also, of course, contain more that one agent, although this is already implied by the properties of communication and co-operation. Additionally, many of these properties are closely linked and may be implicitly created through the provision of others. For example, desires may be simulated using beliefs and intentions (Wooldridge 1993).

The agents in MARINES exhibit most of these features to a greater or lesser extent. The actual details of how these features have been achieved and the level to which they are implemented is discussed later. MARINES system development and communication are discussed in chapter 5. In chapter 7, MARINES agent development is discussed including details of : encapsulation; sensors; beliefs; intelligence; autonomy; planning; and reactive components. In the later chapters, 8, 9, and 10, the results of using these features and future improvements are also discussed.

3.2.3 What is Intelligence ?

The Turing test provides perhaps the most famous description of Artificial Intelligence; if someone is unable to determine whether it is a computer or a human that is responding to their questions then the computer is exhibiting intelligence. However, while it may be a very long term goal to produce a computer program that emulates or even exceeds human intelligence, researchers are generally looking for much simpler intelligence. Van de Velde (1993) suggests that *"it seems fair to call a behaviour intelligent as soon as we, as observers, understand a rationale for it"*. Steels (1993) discusses computer intelligence in some detail and suggests the following as a basis for a theory of intelligence *"the ability of a system to maintain itself through the creation and use of representations"*. In Steel's description, self-maintenance is based upon an agent's ability to decide between two equally valid actions in favour of the one that will keep it, or the system, healthy. For example, a starving worker that is intelligent would choose to eat, rather than work, so that it could live, and work, longer. While Steels does not explicitly state this 'Beliefs', 'Desires' and 'Intentions' are representations that an agent might use to deliberate about its world.

Problem solving ability is also suggested as a prime attribute of intelligent systems. However, Covrigaru and Lindsay (1991) argue that autonomy is more important. Unfortunately, autonomy is also a controversial subject, and this is discussed in the next section.

For this thesis the limited intelligence of a computer can be seen as the ability of a process to be observed to behave rationally, without external assistance, when confronted by a variety of situations that were not pre-programmed. Further, an intelligent process will *"contribute towards the survival of the system ... (and) it is adaptive"* (Steels 1993).

3.2.4 What is autonomy ?

Maruichi et al. (1990) describes autonomy as the ability of an agent to decide on the order in which messages will be processed. That is, a process that is not autonomous will take messages in a pre-defined order and process them. An autonomous process (or agent) will decide which message to process at run-time. A more general description is given by Wooldridge *"Autonomy generally means that an agent operates without direct human (or other) intervention or guidance"* (Wooldridge and Jennings 1994). Castelfranchi (1994) defines a *"heuristic notion of Autonomous Agent"* which includes *"Goals"* and *"Self-Regulation"*. Further, Castelfranchi distinguishes between two styles of autonomy *"executive autonomy"* (similar to that described by Wooldridge above) and *"motivational autonomy"*.

Motivational autonomy is the agent's ability to choose for itself *"it (an agent) is able to make decisions concerning multiple conflicting goals ... it adopts other agent's Goals as a consequence of choice"*. This ability to make choices relies heavily upon Beliefs, Desires and Intentions (BDI) and the argument becomes somewhat circular, is this a definition of autonomy or a definition of what DAI research would like an autonomous agent to be ?

More importantly for this thesis, Castelfranchi suggests that autonomy is a relational concept and in terms of DAI Agents autonomy is restricted ! *"Agents have to be and to act in a way that 'fits' the context."*. The world that the agent lives in determines how it behaves, one agent doing something may induce or prevent another agent from acting as it otherwise would.

Therefore, for the purposes of this thesis 'autonomy' is the ability of an agent to attain its goals without continuous guidance. In doing so it should have regard to other agents actions and environmental changes. It will also have the ability to store and process information in a different order to that in which it is received.

When considering a computer system where an agent or process is not continually guided by an external entity, intelligence and autonomy are closely linked. To be autonomous an agent must also exhibit rational behaviour, and, as soon as it does that Van de Velde (1993), for example, would define it as intelligent.

3.3 Architecture

Studies of architecture for MASs can first be subdivided according to whether they are dealing with the overall architecture of the whole MAS system or the internal architecture of the agents. These sections are entitled *"DAI and MAS system architecture" and "Agent architecture"*, respectively.

Each of these architecture sections may then be further subdivided. As in most agent research, there are a number of conventions for sub-dividing these architectures. Bird, for example, suggests the dimensions of *"distribution, heterogeneity and autonomy"* (Bird 1993) and categorises the MAS according to whether they are loosely or tightly coupled. This would appear to offer a possible framework for a high level taxonomy in the longer term. However, for research purposes many MAS simulate these characteristics on a single processor, or even as a single process (Cohen et al., 1989). This makes it difficult to precisely match these prototype systems to the criteria Bird has provided. Therefore, in this chapter some fundamental architectural differences are described. A knowledge of these options can assist the designer of a new agent system. However, it may be possible to build a similar system using a variety of architectures, or even emulate one architecture using another.

3.3.1 DAI and MAS System Architecture

This section begins by discussing the Multi-Agent Systems that have their foundations in Distributed Artificial Intelligence (O'Hare and Jennings 1996). Steeb et al. (1981) discuss the topological options for DAI architectures used in air fleet control, these are also worth consideration for MAS simulations. Following this, a brief description of blackboard architectures is given. Subsequently, other architectures are considered.

3.3.1.1 Object-Centred Autonomous Architecture (Steeb et al. 1981)

This is a communication free architecture where each agent performs all the situation assessment and planning for its own situation. The history of the situation is stored in order to determine the likely future state. This may be done assuming that the extrapolation of the past will continue unchanged into the future, or by a more sophisticated inference of the changes likely to occur, based upon some model of the developing scenario. When Steeb postulates the use of this architecture for Air Traffic Control (ATC) one suggestion is for *"a complete set of "rules of the road.""* to assist the modelling, this suggestion is important to the MARINES simulation. A further important assertion is that conservative 'cushions' and large resolution lead times are only needed between the aircraft if the agents have incomplete knowledge of the intentions of other aircraft. This is the case in real world ship collision avoidance and shows that this architecture quite closely follows real ship navigation. However, within each ship sub-system a different architecture exists, here the agent in command has control over the assisting agent in a similar way to that described in the hierarchical architecture shown in section 3.3.1.6.

3.3.1.2 Object-Centred Co-operative Architecture (Steeb et al. 1981)

Using this architecture information that might have been gathered by inference is communicated directly between agents. The decision to gather the information by inference or by communication can be made according to the importance of the information and cost of obtaining it; a compromise between communication overheads and processing time overheads. From experience with the use of radio at sea, further consideration has to be given to the veracity of the information; either intentionally or through misunderstanding the agent may obtain inaccurate or incomplete information.

3.3.1.3 Space-Centred Architecture (Steeb et al. 1981)

This architecture is similar to current Air Traffic Control systems, where a control centre makes decisions for all aircraft in a region. This type of control is shown to be very complex, the control centre communicating not only with aircraft but also with other control centres. The control centre must achieve precise heights and directions before handing the control to another control centre or free-flight. The communications between aircraft and control centres must be very robust. However, the system is suited to high capacity traffic movement, if the processing and communication capabilities are high enough; the 'cushions' and large resolution lead times being unnecessary.

3.3.1.4 Function-Centred Architecture (Steeb et al. 1981)

This is similar to the space centred architecture above, however, a processor may control all of one functional aspect of the task within a geographic area. Steeb suggests that one processor in an Air Traffic Control System may control all of the aircraft taking off, or all the ones landing, etc. If the problem space naturally clusters objects into functional groups then this architecture may be suitable.

3.3.1.5 Plan-Centred Architecture (Steeb et al. 1981)

Each processor will attempt to produce a plan for the complete problem from a different perspective. This has advantages if the planning time is long, the solutions are sparse or there is no guarantee that any one method will produce a solution. As soon as one processor comes up with a solution it transmits it, allowing the other processors to abort their solutions. Steeb also suggests this architecture if there are frequent processor losses *"since each processor essentially acts as a redundant element."*

3.3.1.6 Hierarchical Architecture (Steeb et al. 1981)

A hierarchical architecture has supervisory nodes fed with information by lower level nodes. At the top level a decision maker evaluates strategic plans and passes the decisions to lower level nodes that control objects, in this case aircraft. One disadvantage to this structure is the high communication overhead.

3.3.1.7 Blackboard Systems

Several MAS have been built using a blackboard architecture (Maitre and Laasri 1990). In theory, this architecture has a central blackboard upon which many processes can work on a problem and post results. The posted results may then be used by the other processes, each possibly specialised in only part of the task. In fact, blackboard systems for MAS seldom consist of a single, globally accessible, blackboard. Several blackboards may be layered on top of one another; preventing processes from writing results directly, possibly overwriting important information. Alternatively, local scratch pads may be used by the processes and the results passed via a blackboard management system.

Maitre describes blackboard systems as medium grained MAS; each node performs a larger part of a task than, say, a single neuron in a neural network and a smaller part of a task than one, say, using a TouringMachine architecture (Fergusson 1994). Interestingly, the shared blackboards in Maitre's architecture are actually implemented using objects and message passing. This is supported by Hewitt and Leiberman (1984) who suggest that it is not feasible to build a distributed system using a single blackboard and show that message passing can assist when building a system with multiple blackboards.

3.3.1.8 The ARCHON Architecture (O'Hare and Jennings 96)

ARCHON differs from the architectures described by Steeb in that there is inter-agent communication but there is no global control. Agents communicate with acquaintances and volunteer information they believe to be relevant. Each agent can model the other agents of its acquaintance.

In fact ARCHON provides the framework architecture that allows requests for information, requests for processing services and methods of volunteering information. Each agent has an ARCHON Layer, an intelligent system and an interface between them. In the example implementations each agent performs a discrete sub-task; this is somewhat similar to the function-centred architecture. However, there appears to be nothing that constrains an agent to this configuration.

3.3.1.9 Exchanging Data in an MAS System

- Shared Memory

Some Multi-Agent Systems use shared memory to distribute their data (Moffat and Frijida 1994), the main alternative being message passing. Some of the advantages of a shared memory architecture cited are the lower performance overhead of writing information directly and the possibility for new agents to link dynamically, and transparently to the information centre.

- Message Passing

The majority of MAS are message passing systems (Craig 1991). A major advantage is that the security of the information is greater, this is discussed in detail in many texts on Object Orientation (Rumbaugh et al. 1991) and in a cornerstone paper on Information Hiding by Parnas (1972). Importantly, message passing also helps to make agents independent from their environment and hence from each other. If the messaging system supports it, the location of the agent processes also becomes unimportant; whether they run on local or remote machines is transparent to the other agents.

As discussed in chapter 5, there are also Operating Systems considerations for using message passing; protected mode processors may prevent the use of shared memory between most processes. Of course, many message passing methods actually use pointers to shared data in order to pass the messages, the difference between this and a shared memory system being that local copies of the data are then made and all the processes must use the message passing protocols provided. It is not possible for a process to directly overwrite global information.

3.3.2 Agent Architecture

In this section the internal architecture of agents is discussed. Agents may act reactively or deliberatively (plan) or may incorporate a mixture of both (Wooldridge 1994).

3.3.2.1 Reactive Architectures

A purely reactive architecture has no concept of adaptable medium or long term memory; once a decision has been made nothing is stored for later planning procedures. This type of architecture normally responds well in rapidly changing environments (Pollack and Ringuette 1987). A simple real world example would be a child's toy fitted with a 'bump' sensor. When it hits something it reverses, rotates clockwise ninety degrees and sets off again. There is no necessity to stop and consider a plan and no history is maintained. There are two basic flaws to this architecture; firstly, there is a chance of getting stuck in a corner or trying to elude another similar toy or robot (Penders 1993); secondly, a purely reactive agent is unlikely to achieve any complex goals. They have, however, been used successfully for predator/prey simulations (Fenton and Beck 1989).

Note that Fisher and Wooldridge (1993) describe a reactive system in a different way: *"a reactive system is one which cannot adequately be described in terms of 'initial' and 'final' states"*. However, most of the papers on the reactive components of agent architecture concur with the definition as one that responds without explicit reasoning.

3.3.2.2 Architectures for Planning

Architectures for planning, on the other hand, store information and knowledge representations for later use. Examples of knowledge representations are goals, beliefs, desires and intentions. This is often done using a symbolic language, such as Prolog, to assert 'facts'. It is worth noting, however, that these 'facts' actually reflect an agent's current beliefs and some may be conflicting or erroneous. Noise may have affected the agent's sensors, the state of the agent's world may have changed or the agent may have been misled.

Co-operative planning is also an important part of MAS research. Agents may use joint intentions (Jennings 1993) in order to arrive at plans; that is, they may explicitly negotiate and agree upon plans of action. An alternative is that the agents may be offered an incentive to perform a task, as in a contract net (McCabe and Clark 1994). The agents may, on the other hand, infer information from the environment in order to plan without negotiation. This may still be co-operative, benevolent agents may be intentionally programmed to assist each other, some other systems offer a reward. Simulations of insect behaviour have used pheromones to induce co-operative action (Staniford and Paton 1994).

In many MAS the agents include both reactive and planning components (Fergusson 1994), planning to enable goals to be achieved autonomously and reactive to cope with exogenous events (Hanks et al., 1993).

- Production Rules

As has been discussed, beliefs and intentions rely upon the storage of representations. Rules may be used to manipulate these representations. In some systems there may be several sets of these rules. Some of these may be Meta level rules; for example, belief revision rules, intention adoption rules (Wooldridge 1993) and intention or goal achievement rules.

Production rules have the advantage that they may be written and read in almost natural language. This gives users confidence in understanding the logic behind a computer's decisions. Additionally, the rules may be simply manipulated and a domain expert can more easily become directly involved in the creation of a rule set for a specific purpose.

The rules may be parsed in a forward chaining or backward chaining manner. Forward chaining takes the left hand side of each rule in order and matches it with known facts, if all the conditions in the rule are met then it is said to fire. In some systems the rules are prioritised and once a rule fires then the search is abandoned. In other systems several rules may fire and then conflict resolution is performed to choose the appropriate action. Certainty factors may be applied to the rule i.e. IF condition1 AND condition2 THEN there is (0.4) certainty Action1 should be taken. These factors can be used to resolve conflicts.

Backward chaining takes a goal and attempts to find all the rules and hence conditions that could prove it to be true. i.e. if the goal is on the right hand side of the rule then it will take that rule and try to find conditions that satisfy the left hand side.

In complex systems META rules can be used to direct the reasoning. As in a mathematical proof, an adept mathematician will instinctively know that when certain conditions occur together a certain path is more likely to lead to a rapid solution. So, these META rules suggest the most likely way to proceed in order to quickly arrive at a successful conclusion. These META rules are, more often than not, domain specific, however, a few may be included as a general reasoning aids. Such, a general reasoning aid might direct the search to try rules containing the most recently stored facts first.

Whether forward or backward chaining is better or more efficient depends upon the problem to be solved. If the rules are deterministic and can be prioritised into a static list, without too many rules, then forward chaining systems, firing only one rule, can perform well. However, as the number of rules grows this becomes too cumbersome. Then, META rules must be employed and with a large system backward chaining should provide a more rapid solution. Both are essentially top down searches, a complete bottom up search, implied by unordered forward chaining is likely to provide the slowest performance. In any case, if performance is the ultimate goal, there will probably be a more efficient algorithm than an inference engine. The production rule system provides a reasonable compromise between outright performance, flexibility and ease of use.

3.3.2.3 Layered Architectures

Many agent designs split the internal processes into layers. This is often done in hybrid reactive/deliberative agents where a clear distinction is drawn between the reactive and deliberative components. Additional layers may include modelling, co-operation, intentions and learning. These layered designs normally fall into one of two categories either horizontal layered or vertical layered (Muller et al. 1994). There is some discussion about which style should be called horizontal and which should be called vertical. For this thesis the convention proposed by Muller in the description of the InteRRaP model (Muller et al. 1994) is used, as described next.

- Horizontal

Horizontal architectures permit information to pass between the layers. Each layer *"acts as if it alone were controlling the agent"* (Fergusson 1994) and requires a control system to determine which action to perform, as shown in figure 3.1. This has some advantages in allowing multiple concurrent layers to operate independently. However, the architecture can be complex to implement when compared to a vertical one.

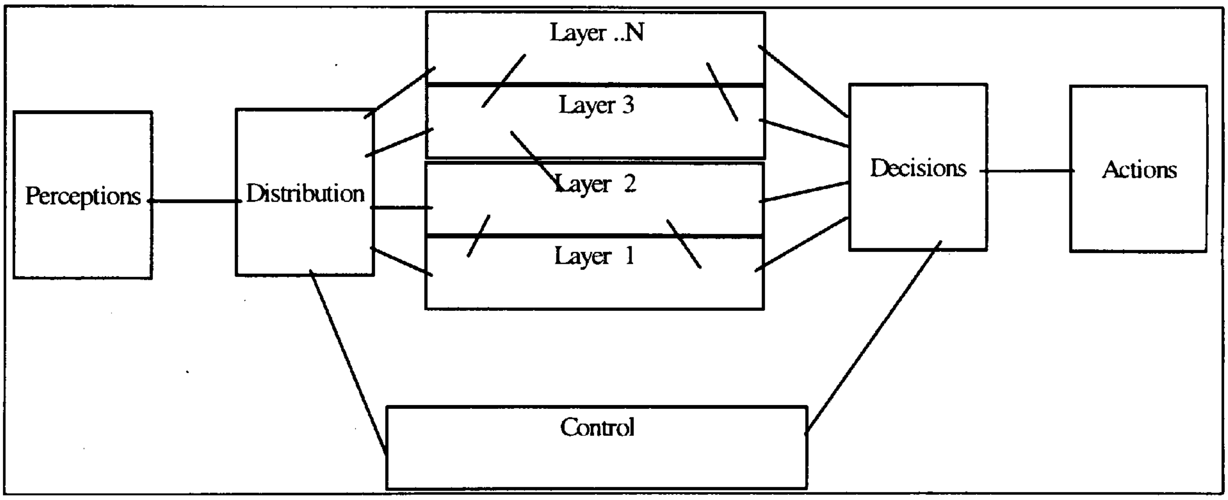


Figure 3-1 A General Horizontal Agent Architecture

- Vertical

In a vertical architecture (Figure 3-2) the incoming information from the sensors is processed first at the lowest level. If the level is unable, or does not need, to process it then the information is passed up a level, and so on, the results being passed back down through the levels and actions taken accordingly. This does away with the need for the external control of the horizontal architecture. However, the implied sequence of processing may possibly slow the calculation of a solution, when compared to a system that calculates all the layers in parallel.

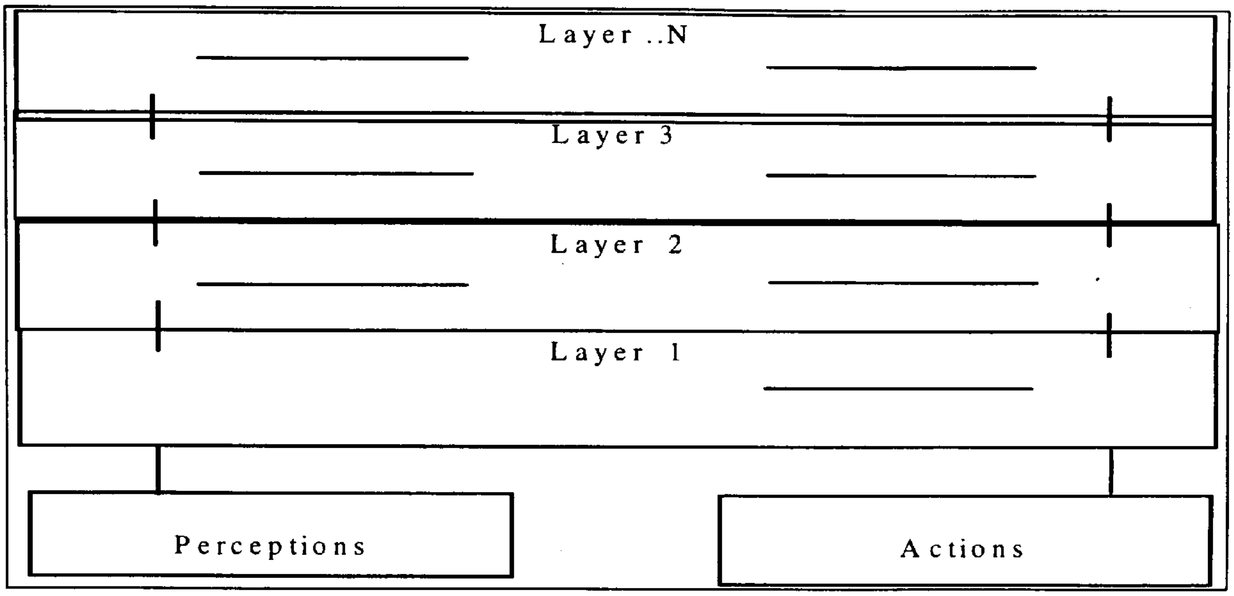


Figure 3-2 A General Vertical Agent Architecture

3.4 Languages and Tool kits

3.4.1 Agent Oriented Programming (Shoham 1993)

Shoham's original paper on Agent Oriented Programming (AOP) has been accepted as a cornerstone for widespread MAS research. As well as Shoham's Agent0 (1991), other research, such as PLACA (Thomas 1994) and MyWorld (Wooldridge 1994b) have taken the paradigm and produced both languages and test beds.

The main thrust of AOP is that agents are modelled on mental states. AOP languages give the programmer the ability to model these states directly. Typically, an AOP language provides messages that follow speech act theory (Searle 1969), informing and requesting as well as statements to manipulate beliefs, desires and intentions.

3.4.2 April (McCabe and Clark 1994)

The April language provides a mechanism for *"the transmission and manipulation of complex symbolic data"*. It is a high level language dealing with the message manipulation required for building distributed systems. Essentially, it allows processes to be defined that can communicate with each other in a uniform manner. A macro language is provided in order to build further layers on top of this message architecture, permitting complex parts of agent programs to be written in other languages, such as 'C'.

3.4.3 PLACA (Thomas 1994)

PLACA is an example of an Agent Oriented Programming language (Shoham 1993). Thomas gives both a BNF description of PLACA and describes an interpreter upon which a PLACA program may be run. There is also a short demonstration PLACA program for an agent that shelves library books and performs Xeroxing tasks. Each agent program would run as a program on an interpreter, which provides communication with other interpreters and hence other agents.

3.5 Agent Based Simulation Environments

Nearly all these changing worlds are designed to study aspects of agent behaviour, not to solve real world problems. They are usually constructed to give an experimenter the ability to test an agent under controlled conditions. A number of parameters may be altered in order to stress the agents. For example, by constraining the time the agent has to do calculations or putting more obstacles in the way of the agent.

The simulations that have been created using agents generally model time in coarse steps, they are not truly continuous simulations (see chapter 2). There are excellent reasons for this in a research context. These systems are normally easier to build (than a fully distributed continuous simulation performing the same task), experiments can run very rapidly, the processor time/world time ratio can be more easily controlled, the experiments may be replicated and the results are repeatable.

While the use of coarse time increments removes some of the problems associated with continuous simulation some of the simulations are quite sophisticated. The agents in Phoenix (Cohen et al. 1989) solve non-trivial problems and assist in the study of complex agent architectures.

The replication of experiments allows researchers to more easily compare their results, even when the same simulation scenario is run at several separate laboratories. Comparisons are easily made between agent performances under carefully monitored and controlled conditions (Hanks et al. 1993). Does an agent with a rapid reactive architecture perform better than one with a more sophisticated but slower planning architecture when the environment changes at a fast rate? Does an agent behave in a predicted manner when it meets another agent?

However, in the real world one is seldom able to exactly replicate experiments, small differences in conditions lead to varied results. By examining the wide range of topics studied in MAS research, and the constraints that have been necessary for those studies to be effected, it can be seen that a MAS can go some way to creating a complex changeable environment if some of the constraints are removed. The realism and variety that distributed systems are capable of is accepted by DAI researchers *"DAI also brings about a new perspective in knowledge representation and problem solving, by providing richer scientific formulations and more realistic representations in practice."* (O'Hare and Jennings 1996). It is further supported by the view of Minsky (1986) and Durfee (Durfee et al. 1989) that it is possible for a number of simple agents together to perform a task that is beyond the capabilities of a single agent.

A changeable environment is especially true in the marine world. Among other things, the current, wind and the personnel operating the vessels are unpredictable. In particular, the navigator of a ship is autonomous and while most navigators will try to obey the steering and sailing rules (IRPCS 1989) they will do so in different ways. It is this continuously changing, non-deterministic environment that the MARINES project hopes to simulate through the use of a MAS. This simulation should not only be rich in variety, but also provide behaviour that goes some way towards mirroring the real world. The simulation should respond in real time to environmental changes and answer some of the criticism of Captain M. Guicharrousse (1990) of simulators that *"respond very accurately to the laws, conditions, artificial and discontinuous situations that the programmer has fed to them"*.

3.5.1 Pengi (Agre and Chapman 1987)

Agre and Chapman explicitly discuss a feature important to most Multi-Agent Systems, robust behaviour under uncertainty. This is achieved in Pengi using '*indexal-functional aspects*'. The description puts forward the argument for using what many later authors describe as 'reactive reasoning'. In Pengi this replaces planning because *'Planning is inherently combinatorially explosive, and so is unlikely to scale up to realistic situations'*. However, in many later systems reactive reasoning is used in addition to planning. This gives the best of both worlds, rapid response to unplanned events in addition to a more considered plan for longer periods. The agents in MARINES use this style of planning/reactive architecture, for this very reason.

3.5.2 TILEWORLD (Pollack and Ringuette 1990)

TILEWORLD is a well known Agent Simulation. The environment consists of a rectangular grid, each square of which can contain a tile, an obstacle, a hole or an agent. Agents are set the task of filling holes with tiles that are strewn around the environment. In most implementations the number, depth and position of the holes changes dynamically, as do the number and position of the tiles and the obstacles the agent encounters. To be successful an agent must completely fill a hole before it disappears as the environment changes. *"The dynamic aspect of a TileWorld domain distinguishes it from many earlier domains that have been used for studying AI planning"* (Pollack and Ringuette 1990)

There have been many implementations of TILEWORLD (Pollack and Ringuette 1990; Kinny and Georgeff 1991; Montgomery and Durfee 1990), some of these have been simplified.

An important feature of the TILEWORLD agents is that they have separate deliberation and planning components. This has implications when an agent consists of several asynchronous processes that run in parallel. It may be possible to determine what immediate action to take quite quickly. However, the planning of future moves may take some time. As can be seen in the section on architecture this separation of components has been quite widely adopted.

As shown in Pengi (Agre and Chapman 1987) the planning can be quite computationally expensive. In TILEWORLD a filtering mechanism is put in place to speed up the planning process by removing unnecessary planning. This is done by analysing the compatibility of the options with the current intentions. If the option is compatible then it passes the filter and is stored for planning. Some incompatible options may also be valuable and a filter override is provided to determine if an incompatible option should be included in the planning.

3.5.3 MICE (Montgomery and Durfee 1990)

The Michigan Intelligent Co-ordination Experiment (MICE) is a flexible test bed for studying and evaluating co-ordination techniques. An interesting division occurs in that Montgomery and Durfee split the research into two types of test bed: test beds that create a fixed environment and study co-ordination techniques; and test beds with a fixed co-ordination technique used in a dynamic environment. MICE has the flexibility to vary both environments and co-ordination techniques. According to Mongomery's criteria, MARINES creates a fixed environment, that of MARINE simulation. That is, although the MARINES environment changes dynamically, it is not possible to easily create an environment for a simulation of another discipline.

It is difficult to determine whether MICE actually fits into this ABSE section or the previous tool kit section. In a way it is a hybrid of the two. It does provide a simulation environment, however, it is rudimentary and can be used to build a wide variety of environments. It has, for example, been used to build simple examples of the TILEWORLD and Phoenix environments.

3.5.4 NTW (Phillips and Bresina 1991)

NTW is similar to TILEWORLD but has no obstacles or holes, the tiles may, however, be pushed around the grid by a wind. An interesting aspect of this simulation is that the agents and the test bed run asynchronously and there is no guarantee that an agent's request will succeed. MARINES also has to overcome this problem, the message system is discussed in some detail in chapter 5.

3.5.5 Truckworld(Hanks et al., 1993)

TRUCKWORLD is intended as a test bed for theories of reactive manoeuvres and to "provide motivating examples for a theory of reasoning about dynamic and uncertain worlds " (Hanks et al. 1993). The use of a wide variety of simulated sensors such as cameras, sonar and radios is of particular relevance to this thesis. A discussion is given as to the different implementations of sensors and effectors in Hanks et al. (1993) paper on architecture.

3.5.6 Phoenix (Cohen et al. 1989)(Cohen and Howe 1988)(Howe et al. 1990)

Phoenix is a fire fighting simulation based in the Yellowstone National Park. The system has three main goals: research into technical issues, such as adaptive planning and scheduling; motivating issues such as *"to understand how complex environments constrain on the design of intelligent agents";* and to provide a commentary on the aims and methods of AI.

The Phoenix environment is quite basic, relying on the agent implementations to supply many of the facilities, such as sensors and effectors. This has the advantage that the system can be used to generate other simulations.

3.5.7 Playground (Fenton and Beck 1989)

Playground is an *"Object Oriented Simulation System with Agent Rules for Children of All Ages"* (Fenton and Beck 1989). The simulation is *"inspired by our intuition that biology provides a good metaphor for understanding complex dynamic systems"* (Fenton and Beck 1989). It has several interesting features, a simple drawing object and phrasal grammar (pseudo natural language) interface, agent rules, a rule compiler, a pull style architecture and facets (classes of properties).

3.5.8 Air Traffic Control (Findler 1991)

Using the simulation of a Distributed Air Traffic Control (DATC) system, Findler discusses the problem of distributed planning. Topics include: how nodes should be interconnected; when should communication occur; how to deal with unexpected events; and how coherent action can be achieved. A co-ordinator-co-worker structure is used. Nodes may take the role of co-ordinators, co-workers or nominators or a combination of the three. The problem selected limits the roles that a node can assume. A nominator nominates the best co-ordinator. A co-ordinator decomposes the problem, distributes the problem to co-workers collects the sub-solutions and synthesises the complete solution. A co-worker will decide which co-ordinator to request the next task from, possibly decompose the problem further and act as a co-ordinator for the further decomposition. This system is said to reduce the communication overhead found in a contract net.

An interesting aspect of the work is the use of distributed scratch pads. These are used to pass information about a task to co-workers, then by the co-workers to solve the sub-problem and then to pass the results back. This allows nodes to recover from errors, once a co-worker is detected as missing the task may then be re-distributed to other nodes.

3.5.9 An Object Oriented Simulation of Autonomous Agents (Craig 1991)

This simulation of agents shows that, in addition to the descriptive model of the simulation, functional modelling of the environment has to occur within the Agent. As the environment changes throughout the simulation the agent's environmental model will need to change dynamically. This environmental model must be sufficient to allow the agent to apply the model to all the situations it is likely to encounter. The simulation further splits simulation objects into three classes *"physical, abstract (or control), and sensors/activators)"* (Craig 1991). Interaction of the physical objects *"can be initiated by mediating spatial phenomena"* (Craig 1991). This is somewhat similar to the agent objects in MARINES, where action is actually stimulated by the clock time but only continues if spatial phenomena, such as proximity of another vessel, make further action essential. However, unlike Craig's system, in MARINES the abstract/control objects normally form a layer between the physical agent and other physical objects such as the memory stores, gauges and MARINES sensors/effectors. This difference is, in part, due to the fact that the MARINES agents are seen as entities in their own right; a fact that is emphasised by them being separate Windows processes. As mentioned above the sensors/effectors in MARINES are modelled on real world objects and act only as receptors for incoming information or transmitters for outgoing information, packaging it as necessary. These perform a similar function to Craig's description. However, there is a clear real-world distinction in MARINES that is less apparent in Craig's system.

3.5.10 My World (Wooldridge 1994b)

Once again this is a grid based environment, this time implemented in Pop-11. MyWorld is designed to create a variety of different scenarios. In the scenario described by Wooldridge the agents are able to move and eat. Co-operation is required between several agents in order to 'eat' resources of high value. A version of the contract net protocol has been used to ensure co-operation; agents bidding for assistance to 'eat' a resource.

MyWorld is based upon the Agent Oriented Programming paradigm (Shoham 1993) in which mental constructs are used to program the agents. As discussed, Beliefs, Desires and Intentions (BDI) are the way that intelligent behaviour may possibly be produced. Wooldridge has chosen beliefs and intentions as the basis for constructing these agents, stating that *"an intermediate 'desire' modality could always be simulated using beliefs and intentions"*. The use of belief revision and intention adoption rules allow an agent in MyWorld to revise its strategy and the use of its private computational resources provide 'cognitive' action.

3.5.11 TouringWorld (Fergusson 1994)

Fergusson developed a discrete event car simulation that is used in understanding agent behavioural ecology. The simulation includes TouringMachines (agents), obstacles, walls, paths, signs and traffic lights. Once again, in a similar way to TileWorld, the environment is created with a number of configurable parameters to enable tests to be performed on agent architectures. The discussion also suggests reasons why modelling other agents' intentions is valuable. For encounters involving only two vessels the intentions of marine collision avoidance are implicit in the Regulations (IRPCS 1989). However, this may become more relevant when multiple ship scenarios are tackled.

3.6 Planning a Multi-Agent System

The basic challenges that must be overcome for distributed co-ordinated problem solvers are described by Gasser and Ishida (1991). Six areas are identified: decomposition of the problem space; communication and interaction; coherent decision making and action taking; how to represent and reason about actions; conflict resolution for disparate agent viewpoints; and how to engineer and constrain practical DAI systems.

Hanks et al. (1993) discuss MAS simulation environments in some detail and put forward some issues that should be considered when planning a test bed for MAS research. Some of these are less relevant when creating a MAS that is not specifically a test bed; repeatability and an easily controlled experimentation environment have to give precedence to the required functionality of the application, in which precise measurements of time and movement are not so crucial. However, the remaining issues appear to provide a suitable starting point when considering any MAS, although from a slightly different perspective. Instead of creating a test bed to highlight these issues, the designer should try to determine how the issues can be resolved in the MAS. Exogenous events will almost certainly occur; the user will change parameters in a way that affects the agents in unexpected ways. The quality and cost of sensing and effecting need to be carefully considered, accurate sensors can supply a large amount of information and need a high communication bandwidth. This point is extremely relevant to the MARINES research; it is surprising how much information is needed to perform collision avoidance. The complexity of the simulated world depends upon careful selection of the components that should be included in the environment and those that should be supplied by the agents. The protocols for communication between multiple agents, determining how they connect, sense, co-operate and disconnect from the environment are crucial to the success of the system. As shown in chapter 5 this includes the loss of messages and the failure of agents. A clean, well defined, interface is essential; the environment should be separated from the agents, for the sake of robustness and flexibility. A well-defined model of time also takes a slightly different meaning. If the simulated time passes at the same rate as world time then any asynchronous processes will require time-stamped information and a notion of the current time. This will give the agent the ability to interpolate or extrapolate environmental changes and plan into the future. It will be noted by the reader that several of Hank's issues are more detailed sub-topics of those Gasser has identified for DAI.

3.6.1 Selecting an architecture for MARINES

Choices have had to be made about the topology of the MARINES architecture. The discussion on architecture by Steeb et al. (1981) was of benefit here. The final choice proved to be a hybrid of the Object-Centred Autonomous Architecture, the Function-Centred Architecture and the Hierarchical Architecture. Much of the reasoning was based upon the real-world environment, following Object Oriented abstraction, however, Steeb's discussions supported the choice. The MARINES architecture tries to obtain lower communications overheads than a truly hierarchical architecture and reflect the natural cluster of navigators, each with a discrete task, within each ship.

The use of separate deliberation and planning components in TileWorld provided useful insight when designing MARINES, as did the use of sensors and effectors in TruckWorld. In particular, the discussion of the Design of Agent Architectures by Hanks et al. (1993) (particularly with the introduction on MAS simulation) was of great value in clarifying decisions during the initial phases of the MARINES project. However, MARINES is intended more for research into how Multi-Agent Systems design may assist simulation than as an aid to agent experimentation. Therefore, the control and feedback is not intended to provide the precise repeatability sought by many agent researchers. In some ways the lack of precise repeatability can be seen as a strength of MARINES; while it may be difficult to obtain definitive results, the simulation produces some unplanned events that are impossible to predict. This is similar to the real world, one person alters their behaviour slightly and many others have to adjust their own accordingly. It is this richness that is considered desirable in a simulation. A student should be able to run the same exercise several times and still be unable to guarantee that a particular manoeuvre for a particular ship will result in safe navigation. Rather, the student will need to remain alert and respond to the developing situation. This should be especially true if more than one student ship is involved in the simulation.

Some of the choices made in the design of MARINES will be considered throughout the later chapters of this thesis. Where relevant, a comparison will be made with the systems described in this chapter. Chapter 5 covers the system architecture, communications and message passing and chapter 7 describes the agent architecture and discusses the 'agency' of these agents. Chapter 10 discusses further research and re-states the most important factors, once again referring to current agent technology.

3.7 Authors Note

In order to constrain this chapter to a reasonable length, this discussion has particularly focused on research that has been influential in the development of MARINES. Although a variety of agent research has been summarised, it still represents a small proportion of a much wider subject. The reader is directed towards "Foundations of Distributed Artificial Intelligence" (O'Hare and Jennings 1996) as a potential starting point for further reading.

3.8 Summary

In this chapter the concepts of 'Multi-Agent Systems, 'agents' and 'agency' have been put into context for the research that follows. This has been done with reference to some of the most relevant literature.

In particular, other MAS architecture and simulations have been described and some of their individual strengths and weaknesses highlighted. Some of the principles and practices used in other MAS have been influential in the selection of the MARINES architecture. In particular the topology of the message framework has been influenced by the discussion of Steeb et al. (1981) and the agents' internal architectures have separate planning and reactive components.

Chapter Four

A Rationale for Simulator Development

4. A Rationale for Simulator Development

4.1 Overview

This chapter expands upon the areas of simulation that are being investigated in MARINES. The discussion includes both weaknesses detected by the author and areas criticised by other users and researchers. Instructor experience, simulator reliability and simulator use are discussed in section 4.3. Then in section 4.4 the realism of current simulators and the need for increased realism are addressed. A number of desirable improvements are derived from the discussion and re-stated in the summary in section 4.5.

4.2 Introduction

Although large simulator installations are still being constructed, such as the Simulation Center at Sydney, Nova Scotia, a number of colleges are opting for micro computer based simulation systems. When building and installing marine simulators based on micro computers it became apparent that such simulators were used less than expected. The instructors criticised the number of errors that the simulators contained. They also criticised the complexity of the software. There is further criticism of unnatural simulator behaviour, and the STCW code (1995) also cites realism as an important factor. Most simulators meet this requirement for simple exercises. However, many are unable to cope with sophisticated exercises involving changeable environmental conditions. A number of desirable improvements have been identified and these are outlined in this chapter.

4.3 Simulator problems

The problems identified and discussed here are largely attributable to the relatively immature micro simulator market. The improvement in the performance/cost ratio of these systems has been mercurial, and the individual processes are becoming more sophisticated. Unfortunately, these improvements have served to highlight some simplifications that were necessary when less powerful equipment was available. Furthermore, the low initial cost and reduced maintenance requirements, of such a simulator, often means that specialist staff are no longer employed.

4.3.1 Instructor training

In training, during micro simulator installations, it became apparent to the author that only a few instructors were familiar with operating large computer systems; even experienced instructors were usually only familiar with one specific operating system. Language was also a problem; while most of the instructors were experienced seafarers and had some knowledge of English, many of the simulators were installed in countries where English was a second language. The normal difficulty of eliciting the users' needs was exacerbated by this language barrier. Together with the technical nature of the manuals, this made a number of simulators almost inoperable by the intended users. Little time or budget was set aside for the training of instructors and the instructors were seldom specialists in simulation. This view of simulator use is supported by the research of Chen (1992).

4.3.2 Simulator reliability

Marine simulators are large complex software systems and Pressman (1992) states that *"testing cannot show the absence of defects, it can only show that defects are present"*. Therefore, as with any large software system they will contain some errors. These errors manifest themselves in a number of ways. The most damaging and difficult to eliminate are infrequent run-time errors that are difficult to reproduce. One of the most difficult the author has had to track down was a conflict between an operating system module, an instructor station process and the network software. This could only be reproduced on the installed version of the simulator, not in a test environment. Even then, it might not occur for more than a week and then happen up to three times in a day. The determination and elimination of the problem took several weeks and eventually the operating system was replaced with a different version. The original version of the operating system and network software were used throughout the remainder of the simulator without a similar problem, probably because little, interrupt driven, keyboard I/O was handled elsewhere.

If this type of error terminates the simulation, perhaps more than an hour into a complex exercise, it is very frustrating and also damaging to the students' attitude. It can also take several months without a failure to restore the users' confidence. If the simulation can be permitted to continue, albeit with less functionality, or even terminated in a less abrupt manner this would be advantageous.

4.3.3 Simulator use

Companies such as Maritime Dynamics, MicroSim, Transas and PC Maritime are all supplying distributed micro computer based products. These products are more affordable and are now being installed in relatively small colleges. In many cases, the limited need for simulation does not warrant the employment of a full time simulator instructor. Therefore, the teachers perform the simulation as a part of their daily duties. The employment of specialist, full time, on site technical support staff has also, by and large, been discontinued.

The fragile nature of many simulators intimidates the instructors. Therefore, unless an instructor has a special interest in simulation, the simulators are seldom used. While there was no way of guaranteeing the clock accuracy, less than two hundred hours of use was shown on a three year old simulator upon replacement. Certainly, the old simulator was never used during the several months spent installing the new system.

4.4 Realism

Realism can be split into two general categories. Firstly, the quality of the components in the simulator. For example, the accuracy of facsimile equipment and the standard of the 3D graphics images projected. This has been investigated by Kim(1990) for port design simulators, he found that, although desirable, a highly accurate facsimile was not essential. Furthermore, Chen (1992) observes that "to effectively train the development of many ship manoeuvring skills a very high level of fidelity is not usually required in a visual scene". The second category, the accuracy and realism of the simulation compared to the real world behaviour was considered more important. For example, do the manoeuvring characteristics of a simulated ship match those of the real ship it is based upon? It is this second category that is being addressed in MARINES.

4.4.1 Realistic target ship motion

On a conventional simulator it is difficult for the instructor to plan a scenario that contains a large number of target ships, particularly when the exercise extends over a lengthy period. The target ships have to interact with the own ship in a believable manner and, if the students manoeuvre in an unexpected way, pre-planned target courses and speeds may not highlight the desired problems. Therefore, realistic target ship motion and interactive manoeuvring has been limited in many simulators, normally in favour of rapid response and ease of use. The performance of the computers has also been a factor; simple target ship motion takes less processing power and less programming effort. Many valuable exercises can be designed taking these limitations into account. However, the processor power is increasing, the graphics of the visual scenes are becoming more realistic and more complex exercises can be supported. A larger number of computer generated ships will create a greater opportunity for interaction and it will become more difficult to plan exercises completely before they commence. Improved support for static exercise planning is already offered on newer simulators, such as, the Transas Ship Handling Simulator. However, in the real world complex dynamic changes occur. As simulators become more realistic, strategies will have to put in place to cope with the dynamically changing environment.

Ship handling, collision avoidance and track keeping are complex subjects; unpredictable forces, such as weather and current, play a large part in determining ship behaviour. The complexity is exacerbated by the human navigators aboard the ships, each navigator having their own characteristics. Guicharrousse (1990) suggested that, using simulators, it was possible to train students in one particular way, that for a particular situation, would provide a good solution on a simulator every time. However, when such a situation occurred in the real world, this might, or might not, be the proper action to take. This was also identified during MARINES evaluation by one of the experienced interviewees, this is discussed in chapter 9. Therefore, Guicharrousse's view is that, while experience can be augmented by simulation, simulation should certainly not be seen as a substitute for real world experience. For example, in training ships' officers and pilots, the requirements for periods spent at sea should not be reduced because of simulator courses. A major reason behind Guicharrousse's criticism is that a computer is *"a perfectly stupid object; it only does what it has been taught to do"*. Therefore, simulators do not realistically represent the complex environment of the real world; where a large number of factors go towards producing a unique situation in every case. In the context of Guicharrousse's argument this is a valid criticism, one that is implicitly accepted in the design of many simulator exercises; such exercises often contain only a small number of vessels and are designed to recreate a particular situation precisely. The present genre of simulators are valuable tools for demonstrating specific problems but lack the ability to respond dynamically to the environmental changes found in a complex world.

Further, the STCW code (1995) defines performance standards for simulators that are used for training and assessment of competence. For example, a training simulator must *"have sufficient behavioural realism to allow a candidate to exhibit the skills appropriate to the training objectives"*. Therefore, for more advanced exercises more realism is required. For example, in exercises that incorporate the effects of land and sea stabilised motion, all the simulated ships should manoeuvre realistically and be affected by the set and drift of the current.

4.4.1.1 Sea stabilised motion

The effects of land and sea stabilised displays for electronic navigational aids has been debated in a number of papers. Smeaton (Smeaton et al. 1994) shows the effect on target histories and vectors on an Electronic Chart Display (ECDIS) and discuss the advantages and disadvantages.

What does not seem to be widely considered is the effect of having a sea stabilised own ship and land stabilised target ships on a simulator. Consider the two extremes in Figure 4 - 1, with two ten knot ships meeting head on, encountering a three knot current. When the own ship is travelling with the current the target ship appears to be travelling at thirteen knots. When travelling against the current the target will appear to be travelling at seven knots. If both ships were sea stabilised the approach speed would be ten knots in both cases. If a simulated ARPA is being used then this information will be displayed on the own ship bridge. The information that is normally displayed on the simulated ARPA accurately depicts what is happening on the simulator, it is the target ship motion that is at fault.

Now consider the effect when a cross current applies as shown in Figure 4 - 2 and Figure 4 - 3. In the example on the left the sea stabilised own ship is set northwards by the current and is involved in a close quarters situation. A simulated ARPA would detect that the ship was crossing and display a crossing vector, however, from its visual aspect the ship would appear to be passing clear. In the real world both ships would be affected by the current and be set northwards, passing clear of each other, as shown in the right hand diagram. Furthermore, in the real world an ARPA would display the target ship on a reciprocal course.

If the ships are moving relatively slowly compared to the rate of the current, the use of land stabilised target ships will result in manoeuvres that fail to achieve the desired passing distances and confuse rather than instruct the student. Exercises are performed in chapter 10 that further demonstrate this effect.

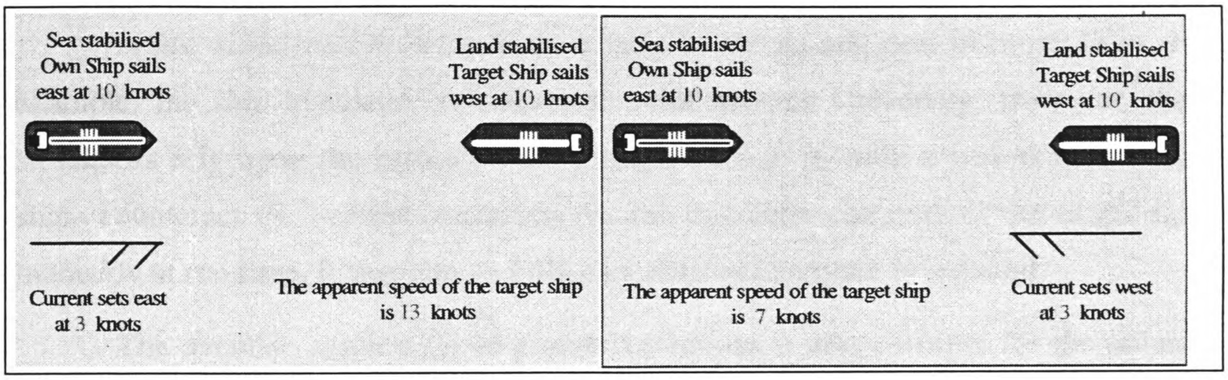


Figure 4 - 1 The effect of current on approach speed on a simulator with sea stabilised own ship and land stabilised target ships.

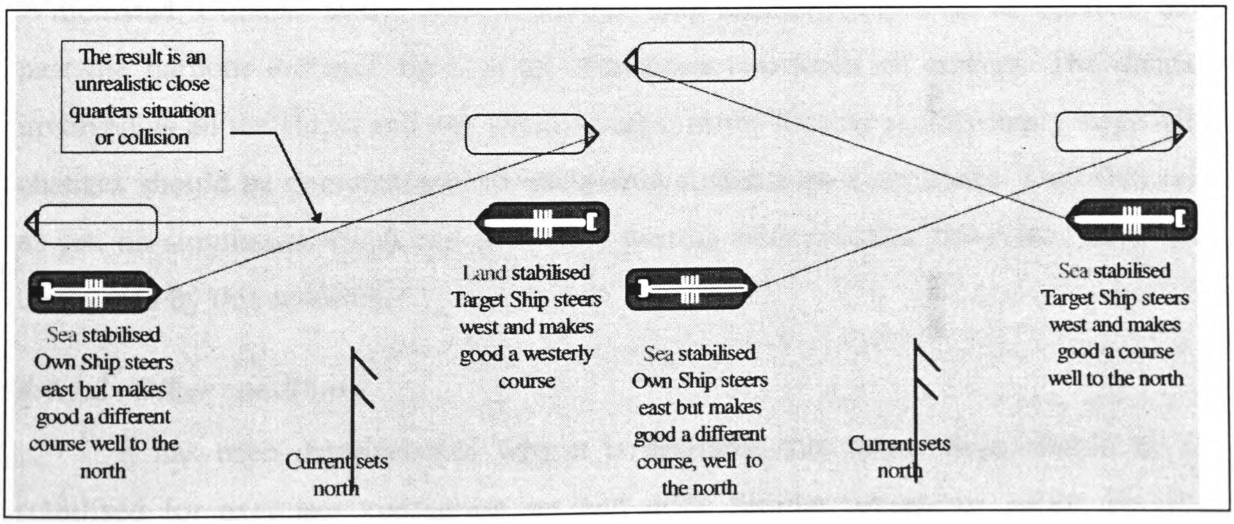


Figure 4 - 2 The effect of a cross current in a nearly head-on situation on a simulator.

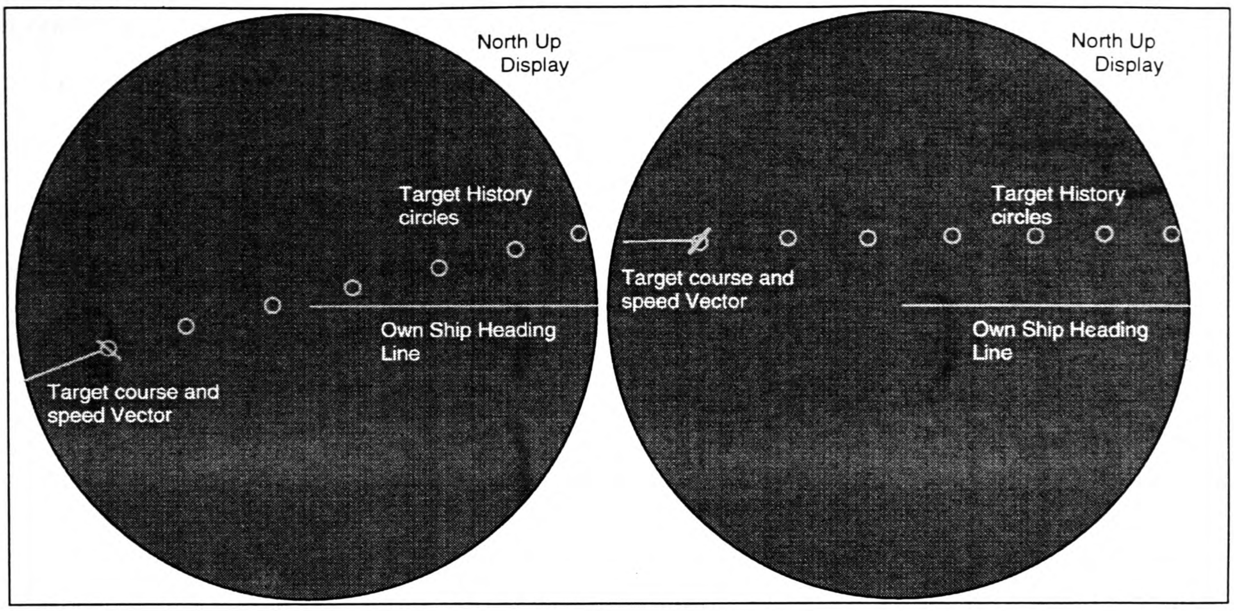


Figure 4 - 3 The radar plot for Figure 4 - 2 from the own ship bridge

A few simulators do include the effects of the set and drift of target ships. For example, the ship simulator at Liverpool, John Moores University. However, these simulators rely upon the instructor designing exercises in such a way that the target ships counteract the current. Alternatively, the instructor can control the target ships manually at run time. Either way, a fully experienced instructor is required.

The dynamic application of course corrections is also essential for the accurate simulation of a target ship encountering changing current conditions. This is particularly noticeable when entering a port or channel. When inside ports such as Willemstad, Curacao in the West Indies the ship encounters little or no current, once past the harbour entrance there is an immediate two knots of current. The dangers involved as all the ships, and any attached tugs, in the locality rapidly apply large helm changes should be demonstrated to navigation students on a simulator. Unfortunately, as yet, no simulators which can accurately portray such complex behaviour have been uncovered by this research.

4.4.1.2 Other conditions

It has been demonstrated why it is desirable that target ships should be sea stabilised for exercises containing set and drift. Similar arguments apply for other conditions such as wind, shallow water effect, bank effect and transverse propeller thrust. These are not considered in this thesis.

Exercises involving, faulty own ship speed, land, stranded or anchored vessels and vessels carrying a different amount of leeway would exhibit some of the characteristics shown for a land stabilised ship. It is essential, however, that the simulator is as accurate as possible for the specified environmental conditions, so that specific problems can be demonstrated to the students.

4.5 Summary

Simulators based upon micro-computers are becoming affordable for a wide variety of tasks. The instructors of such systems, while skilled in other areas, often have little computing experience. The exercises should be believable and new exercises should be easy to create. Novice instructors should not be intimidated by the initial complexity of the system. The basic functions such as switching on, selecting an exercise, starting and running the simulation should be easy to perform. Therefore, to be an effective tool, a simulator should be highly realistic, robust and easy to use and additional functionality should be available, allowing more advanced instructors to perform more complex simulations.

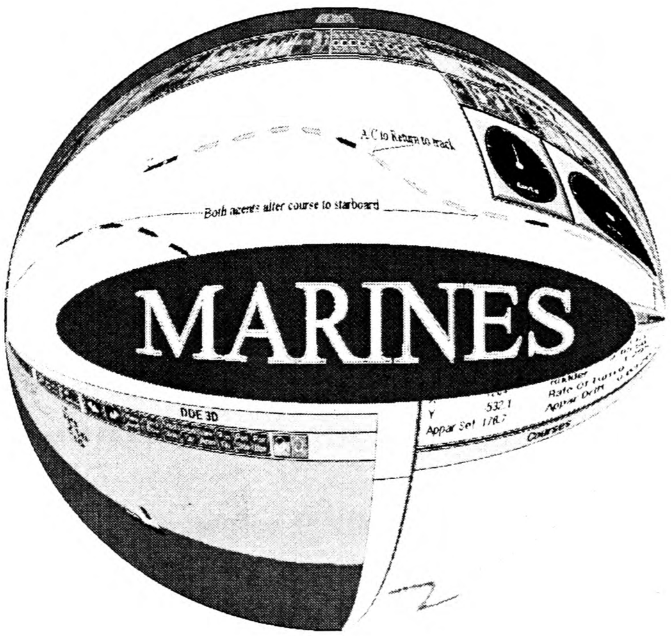
Unfortunately, this is not the general case. As with any large software system, simulators contain errors. These may manifest themselves as occasional glitches or may even terminate the simulation. Simulators often contain simplifications, some of which lead to unrealistic effects. The errors and complexity also make the simulators hard to control and make instructors apprehensive.

When a new simulator is installed or an existing simulator upgraded there are often latent defects that cause occasional errors in the simulation. Using current testing methods, it is impossible to prove the absence of errors in such large software systems. This is particularly true of continuous simulations where it is very difficult to precisely repeat simulations and, hence, to replicate error situations. However, it is possible to show that software is able to tolerate certain types of errors. Total recovery may not always be possible but damage limitation may permit the simulation to continue.

In order to make a marine simulation believable the computer generated ships should have similar handling characteristics to real ships and normally manoeuvre following the collision regulations. At the moment, these manoeuvres are usually provided by the instructor and considerable pre-planning is needed to ensure that computer generated ships do not collide. The ships should also be affected by natural forces such as the set and drift of the current and the dynamic changes found in a complex world. At the moment, a few simulators offer either reasonably realistic ship motion for a limited number of ships, or rule based collision avoidance. One of the purposes of this investigation is to bring research on automatic track keeping and automatic collision avoidance together to determine if this can produce a simulation giving reasonably realistic target ship motion and automatic collision avoidance. This should also provide a mechanism whereby the instructor need not monitor every ship in the simulation all the time.

In the real world each navigator will have slightly different characteristics. When two navigators work together, these minor differences will create a unique solution to a developing situation. A MAS system reflects this and it is hoped that minor changes to an agents beliefs will also contribute to a unique and complex simulation.

Chapter Five



Introduction To MARINES

5. Introduction To MARINES

5.1 Overview

This chapter describes the MARINES system (Moon and Tudhope 1995a, 1995b, Reddy and Moon 1995). Firstly, there is a conceptual description of the MARINES system in section 5.2. Then a general description of the software system is given. Next, in sections 5.4..5.8, Object Orientation, message passing vs. shared memory, Microsoft Windows, Dynamic Data Exchange and C++ are briefly described and the reasons for selecting these features for the system implementation are given. After this, the message passing framework of the system is described in some detail in section 5.9; the challenges that have been met, the solutions arrived at and the changes that have been made as the research has progressed. In reading, one should note the similarity between the agent connections in the conceptual view and in the information layer of the message framework; the information layer is the implementation of the communications in the conceptual system.

In section 5.10 a brief comparison is made between the alternative architectures described by Steeb et al. (1981) and the MARINES architecture.

5.2 A conceptual overview of the MARINES system

The MARINES Multi-Agent System (MAS) was originally conceived in 1993 and is designed to simulate a bridge simulator instructor station. A complete rationale for the use of a MAS in simulation is given in chapter 4. Essentially, it was hoped that a robust, flexible simulation with some assistance for the novice instructor could be developed. Whether this has been achieved is discussed in chapter 9.

At a conceptual level the MARINES environment forms a dynamic world for the agents and other processes to attach to. An interface is provided to give the processes the ability to request information about the environment and adjust parameters within the environment. The intention is that the interface should permit the passage of information that closely resembles the information supplied by the real sensors and delivered to the real controls used to control a ship.

The processes connected to the environment act autonomously to navigate the ships and are called autonomous agents. Several instances of the same agent may be connected to the environment and/or mixed with instances of different agents. In some cases, a different agent may replace an existing one, changing the characteristics of the simulation. Figure 5.1 shows a conceptual view of this system.

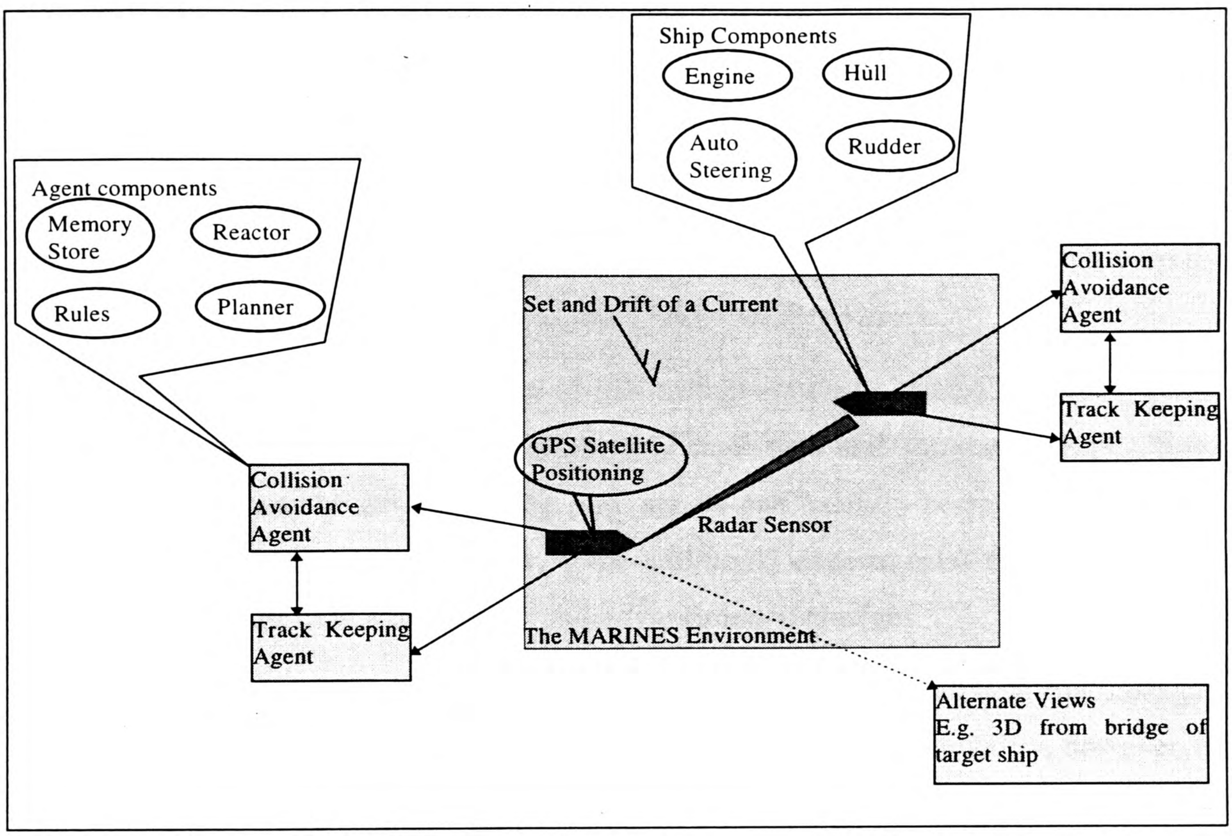


Figure 5-1 A Conceptual View of the MARINES system

Ideally, the central environment will provide the critical functionality of the simulation and the additional agent processes will provide enhancements. Therefore, the failure of an agent process need not lead to the termination of the simulation.

5.3 A description of the MARINES software

MARINES has been developed as a research vehicle. In order to accelerate the development, some of the features that are desirable in a commercial simulator instructor station were not included at the outset of the project, and the implementation of some other features was quite rudimentary. Functionality has been added as the focus of the research has dictated the need. The features that were needed are outlined in

Sections 5.3.1 and 5.3.2. No doubt, future developments will require more complete coverage of normal instructor station functions.

The **M**ulti-**A**gent **R**ealm for **I**nvestigating **N**avigators' **E**ducational **S**imulators (MARINES) is a cross between a generic instructor station for a marine bridge simulator and a test environment for a Multi-Agent System. The instructor environment process forms a dynamically changing environment (Pollack and Ringuette 1990) for the autonomous agents. These agents then attach to the environment using Dynamic Data Exchange, inferring the environmental changes through sensors and changing the ship parameters through effectors that simulate the ship controls. A 3D process is attached using DDE that shows the view from the bridge of the active ship, this permits the instructor to visually compare the aspect of approaching vessels.

The rationale for the selection of this hybrid system is twofold. Firstly, unlike most MAS test beds, MARINES is reasonably close to a real simulation application. This goes some way to demonstrating how agents can usefully perform a real world task in a new subject area. Secondly, some additional experimental control has been added to permit fast time experiments and environmental changes.

The use of multiple asynchronous processes, and an environment that changes continuously results in some loss of experimental control and repeatability, however, it more accurately depicts the real world. In a real application functionality, accurate portrayal of complex behaviour, rapid response to user input, robustness and ease of use all play a part.

5.3.1 The environment

The central environment, shown in Figure 5-2, is the process that provides a sub-set of the functionality found in an instructor station on a simulator. A plan view of the scenario is provided, permitting the relative position and motion of the ships to be viewed. In version 1.07 of MARINES ten computer generated target ships are provided, one super tanker model, one fishing vessel model and eight tanker models. The performance of the models is discussed in chapter 8. A small piece of test land is included, although this does not constrain the computer generated ships. The land was included simply for aesthetic reasons because several of the users commented upon the omission. The historical tracks of the ships can be displayed, showing how the ships respond to changing conditions, such as changes in the current and the dangers presented by other ships.

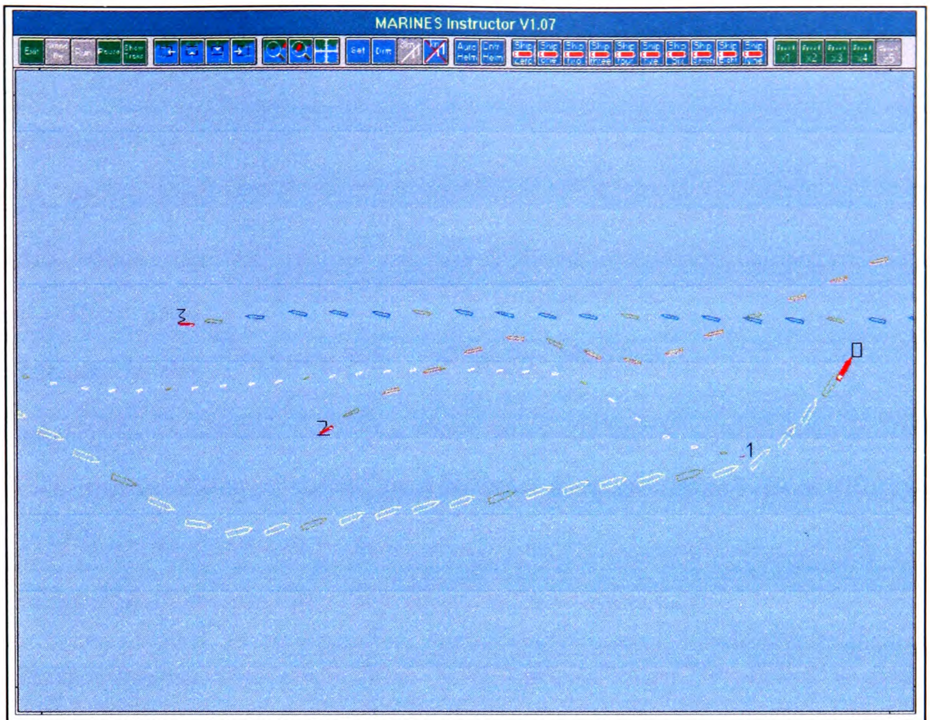


Figure 5-2 MARINES Instructor environment

Some controls that are common to many simulator instructor stations are provided: controls to pan and zoom the plan view; controls to start stop, pause and exit the simulation; controls to enter the set and drift of a current; and controls to select the computer generated target ships; a control to display the historical tracks of the ships.

Additional controls have been added to assist with the research experiments. There are controls to set the simulation rate between world time and five times world time; this permits experiments to be performed more rapidly. Further controls permit the current effects to be enabled or disabled for any ship, thus allowing comparisons to be made between sea stabilised and land stabilised motion.

An interface is provided for DDE connections to the environment. The interface permits requests for obtaining information and issuing orders. This follows the paradigm set out by Shoham (1993) for Agent Oriented Programming, based upon speech act theory (Searle 1969), for requesting, informing and ordering, examples are: obtaining own ship information, such as speed, heading, position, etc.; information about visible targets, such as range, bearing and aspect; manoeuvring instructions, such as setting the telegraph and the desired auto pilot heading. In this prototype version up to thirty processes can be connected to the environment at any one time.

5.3.2 The 3D view

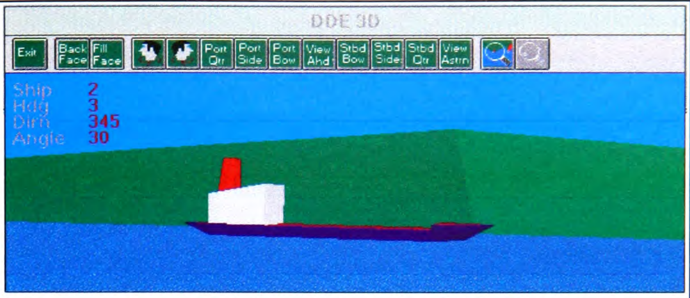


Figure 5-3 Simple 3D view

One process that is connected via this DDE interface provides a 3D view from the bridge of a specific ship, as shown in Figure 5-3. This process is not autonomous in the manner of an agent described in chapter 3, it is a view from an agent's perspective. This permits the aspect of an approaching vessel to be assessed, demonstrating the difference between the aspect of ships under land stabilised motion and sea stabilised motion. The height of eye of the view depends upon the ship that has been selected.

It is hoped, in the future, that additional 'intelligence' may be added to make this process assist the user in tracking the most relevant target. Further views may also be provided that simulate a specific sensor type.

5.3.3 The agents

At the time of writing two general types of agent have been created. Firstly, an agent type that performs collision avoidance between the ships following the Regulations for Preventing Collisions at Sea. Secondly, an agent type that performs track keeping, following the desired courses set by the instructor.

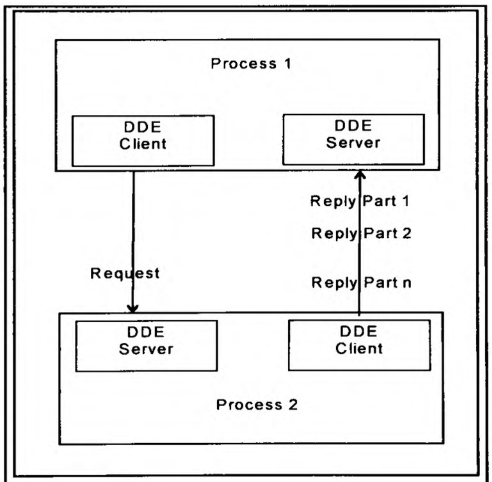


Figure 5-4 DDE Connectivity

The agents connect to the DDE interface provided by the instructor environment. Each process in MARINES is both a DDE client and a DDE server, messages are poked to the servers by the clients. Once connected to the environment server the agent requests that the environment connects back to the agent's server. A typical message sequence can be seen in Figure 5-4, process 1 sends a request for information, process 2 can send the reply in several parts, each message delivery being performed asynchronously.

In automatic mode, each collision avoidance agent applies rules based upon the International Regulations for Preventing Collisions at Sea in order to avoid approaching vessels. As each agent is an independent entity, the actual rules used can be varied for each instance of an agent, thus, an agent can be modelled that misinterprets a particular rule.

Each collision avoidance, shown in Figure 5-5 agent performs collision avoidance in much the same way as a human navigator. The agent sends the environment a request for visible target ships, a message is returned to the agent for each ship within visible range. The information is stored for a number of iterations. When sufficient information is available, the historical positions are extrapolated to infer whether a dangerous situation is developing. This is based upon the CPA and TCPA of approaching targets, a formula for the danger coefficient (Smeaton and Coenen 1990) being used to determine which target presents the greatest danger. If such danger is detected then the rules are parsed to determine what, if any, action should be taken to avoid the collision.

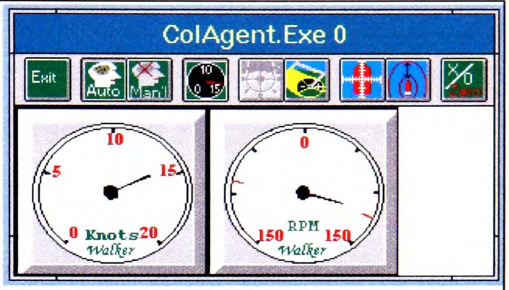


Figure 5-5
Collision avoidance agent

For each agent controlled ship, a track keeping agent, shown in Figure 5-6 co-operates with the relevant collision avoidance agent. The waypoints for the desired track are entered by the instructor/experimenter, then these agents adjust the course of the ship to try to keep the ship on the desired track. The set and drift of the current is calculated by the agent based upon a comparison between the dead reckoning position and the actual course made good. Each track keeping agent also has beliefs about the manoeuvring characteristics of the ship it is controlling, so that it can anticipate when to begin an alteration of course.

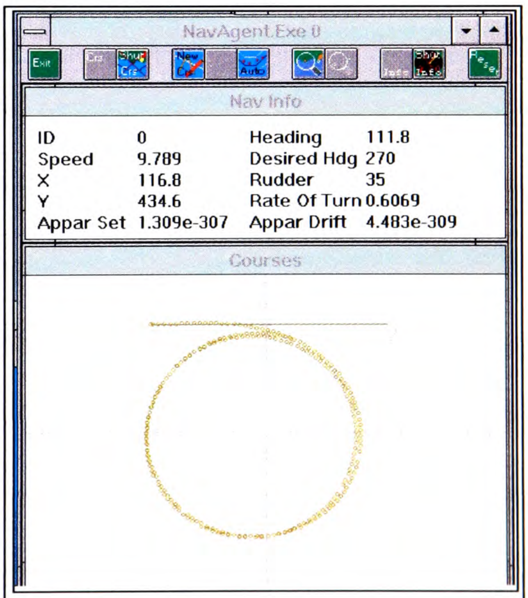


Figure 5-6 Track Keeping Agent

5.4 Object Orientation

Agents and objects are closely related. Opinions differ widely on what makes an object an agent, and this has been discussed in chapter 3. However, there is widespread agreement that autonomous agents can be considered as objects. All the agents will have similar components, planning, reactive and communication components. Additionally, marine simulation contains many instances of real world objects. Ships are obvious examples; each ship is built from similar components, hull, engine, gauges, etc. Thus, in concept, Object Oriented Analysis, Design and Programming appeared to be well suited for the task of building a marine MAS.

5.5 Message Passing vs. Shared Memory

Two options were considered for the interface between the agents and between the agents and the environment, either a message passing (Craig 1991) or a shared memory architecture (Moffat and Frijida 1994).

Object Orientation relies heavily on a message passing paradigm. Such a system enables data hiding (Parnas 1972) and has been shown to enhance robustness and simplify maintenance (Pressman 1992). A further advantage of message passing is that the destination of the messages can be transparent; the sending process can be unaware of whether the message is sent to a process on the same machine or a remote one. The main disadvantage of a message system is the overhead of encoding, decoding and transmitting the messages.

Shared memory on the other hand provides direct and, therefore, normally faster access to the data. However, data corruption can occur if some form of write locking has not been implemented. To prevent this Moffat and Frijida (1994) only give processes a limited ability to access shared memory, improving the mix of performance and robustness for MAS use. The problem has also been tackled in several blackboard systems; local knowledge sources and layered blackboards have been suggested (Jackson 1990).

A further problem with using shared memory is that most multi-tasking operating systems employ virtual memory management. Shared memory, however, must reside at a fixed memory address, or have a handle that can be de-referenced in order to obtain the memory location. This can have implications on the performance of the virtual memory system, possibly leading to disk thrashing. In any case, the agents in a fully operational Multi-Agent System will normally be distributed and even on the same machine most application processes run in a protected processor mode. Therefore, it may become difficult, or even impossible for application programs to share memory (Myers and Doner 1992).

For these reasons message passing has been chosen for the implementation of MARINES.

5.6 Microsoft Windows V3.1

Other research MAS, such as Phoenix (Cohen et al., 1989) and Tileworld (Pollack and Ringuette 1990), have been simulated using a single process and coarse variable time steps. This has been done for several reasons: simplification of the implementation; repeatability of experiments; accelerated performance; etc.

However, to create a MAS for continuous simulation, with rapid response to user input, multi-tasking capability was seen as being a pre-requisite; allowing the multiple agent processes to run pseudo-concurrently. True multi-processor, multi-threaded programs were ruled out because of programming complexity and hardware availability; at the outset of the project development tools for more advanced 32 bit operating systems, such as Windows NT, or even Windows '95, were not available to the author.

Microsoft Windows 3.1 is a co-operative multi-tasking operating environment where the applications are driven by Windows messages arriving in their message queue. Dynamic Data Exchange is supported by Windows, allowing inter-process message passing. These features and widespread availability were influential when selecting Windows 3.1 as the development environment for MARINES.

Borland's Object Windows Library (OWL) provides an Object Oriented Application Framework for producing Microsoft Windows programs. This framework also provides each application, and each window, with an IdleLoop; this allows background processing to take place when no messages are waiting in the message queue.

Therefore, in MARINES the processes are driven by events, such as, instructor interaction, DDE messages from other processes and the IdleLoop routines. Furthermore, under Windows the order of selecting a process to use the processor is not guaranteed. These features emulate a scheduled multi-tasking system. This gives the impression of a continuously changing environment, although, in fact, each process takes short periods of processor time and must release the processor when it is finished. To perform a continuous simulation on such a system the software has to be designed to perform tasks in small increments. Even so, the time slicing is coarser than a scheduled multi-tasking system and one errant process can take over the processor, preventing other processes from updating. Windows 3.1 does, however, provide a <Cntrl> <Alt> <Delete> interrupt facility that can terminate a process that misbehaves in this way, normally, without affecting the other processes. This improved robustness, over Windows 3.0 was also influential in the selection of Windows 3.1 for MARINES.

5.7 Dynamic Data Exchange

Dynamic Data Exchange (DDE) is supplied with Microsoft Windows v3.1 and allows blocks of data to be passed between Windows processes using a client/server architecture.

A single DDE message actually consists of a number of Windows transactions (Norton and Yao 1992). Therefore, it is possible for processes to pre-empt each other and DDE messages may fail to be transmitted. The order of DDE messages arriving is not guaranteed, although, thus far, they appear to arrive in order.

DDE advise loops can be used to enable the server to update a client as and when data changes. This has not been used because of the lack of flexibility and the amount of unnecessary data that could be transmitted to an agent. For example, an agent might need some data frequently and other data infrequently, it would be difficult to provide precisely the correct data at the required times using an advise loop.

In fact, in MARINES, the Microsoft DDEML dynamic link library has been used to provide the DDE connectivity. The DDEML library provides a wrapper for DDE functionality at a higher level of abstraction than the basic DDE commands, making conversation management easier and error checking more robust (Myers and

Doner 92). Myers also suggests that shared memory will not be available in future Windows versions. DDEML and OLE will, however, continue to be supported.

In this prototype system all the DDE message blocks are the same size, making it easy to send and retrieve the data. This solution was chosen to make implementation easier for this prototype. It is possible, and would be more efficient, to transmit only the minimum amount of data required by a specific message.

5.8 C++

C++ is widely used for commercial system development. The reasons are manifold: efficient object code, graphics support libraries, math support libraries, object support, hardware accessibility, wide availability of high quality development tools, etc.

Experimental programming (Sommerville 1995) may require numerous changes and re-configuration of the program. The lack of support for tracking run-time errors is a weakness when using C++ for experimental programming. This is very noticeable when compared to interpreted languages such as Visual BASIC. In a continuous simulation it can be difficult, or even impossible, to precisely recreate the circumstances leading to an error. Debuggers such as Borland's Turbo debugger and Microsoft's code view have improved the situation a great deal. However, they do not support debugging for multiple instances of a program running simultaneously. However, under Windows 3.1 tools such as Winspector, Dr Watson, Winsight, Spy, DDE Spy and Heap Walker assist in this area.

At the inception of the project Borland's Delphi was an unknown quantity, Borland Pascal 7.0 included some Object Oriented features although not as many as C++. Released soon after the project began, Delphi's strengths only really became apparent several months later. If the languages were re-evaluated no doubt Delphi would feature as a strong rival. The efficient code compilation and high productivity would be strong recommendations. One reservation lies with the non-standard language implementation, supported by only a single vendor.

After choosing Microsoft Windows 3.1 in 1993 the only main stream, compiled, Object Oriented languages available to the author at a competitive price were C++ and

Pascal. Overall, C++ was strongly supported and the author's familiarity with Borland compilers led to the use of Borland C++.

5.9 The MARINES Message Framework

5.9.1 The original message framework used in MARINES

Version 1.0 of the MARINES system contained only two collision avoidance agents and two ships, no track-keeping agents were available in this prototype. Each agent acted as the officer of the watch for one of the simulated target ships and performed collision avoidance manoeuvres if the other ship approached. The agent sent messages to the environment requesting "visible" target ships. In this case only one target could be visible, the one controlled by the other agent.

For the purposes of this discussion the framework can be considered as having two distinct layers. The transmission layer using DDE and, at a higher level, the information layer showing the actual content of the messages and to which processes the messages are delivered. There is of course a Windows implementation level for DDE using atom tables and shared memory, but this is not described here.

5.9.1.1 The transmission layer

The original message passing framework was kept as simple as possible. The environment was able to connect to a maximum of four agents and each agent only communicated with the MARINES environment; they did not communicate with each other. No procedures were in place to handle recovery from message transmission failures or a run-time error in one of the communicating processes.

A relatively long time-out period seemed to prevent messages from being lost. If a second, or subsequent, message arrived while a first was being processed the DDEML library seemed able to cope, creating a new stack frame for each incoming message.

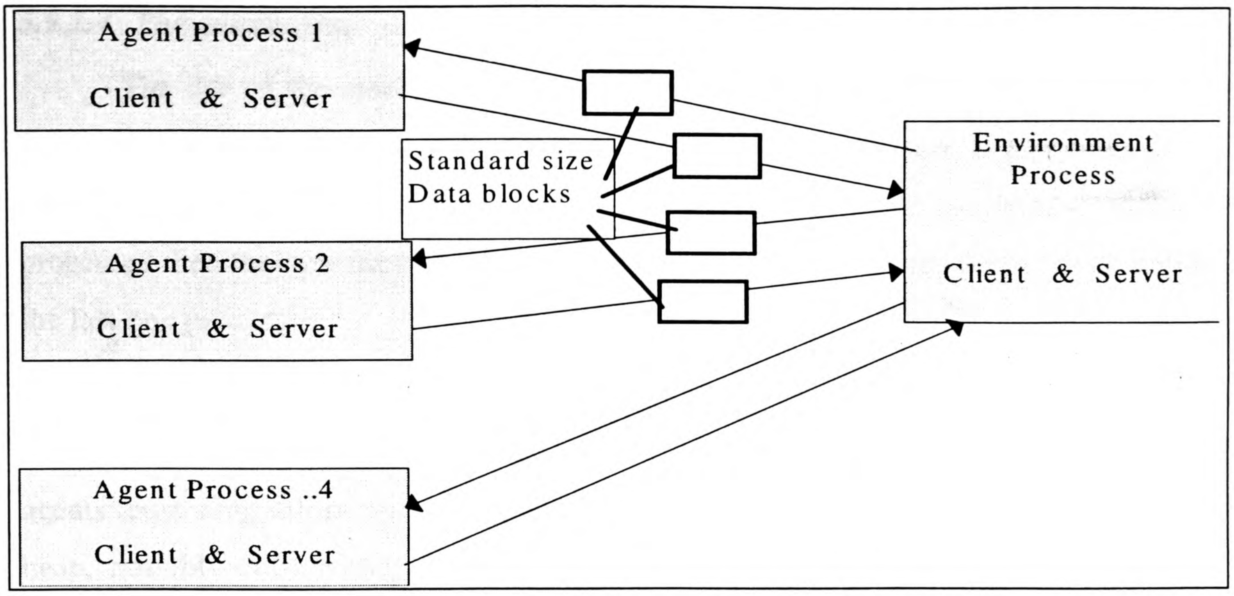


Figure 5-7 The transmission layer architecture V1.0

5.9.1.2 Message content, the information layer

The actual information contained in the messages is an abstraction of the information that a human navigator would use to plan collision avoidance and track keeping manoeuvres. Messages containing new settings are also available allowing the agent to adjust the controls of the ship. Each message is designed to reflect a specific type of ship control or sensor.

For example, in order to determine the danger of collision between the ships, the range and bearing of each visible target can be retrieved by sending a 'look' into the environment. This sensor is equivalent to the information available to an officer detecting a target by radar.

5.9.1.3 The protocols

The initial version of MARINES used a mixture of remote invocation and synchronous message passing (Burns and Davies 1993), depending upon the type of message being delivered. All incoming messages were decoded immediately. Each message was processed as it arrived and any necessary replies were sent before returning a successful transaction flag to the sender.

5.9.1.4 The constraints

The use of the stack means that the order of processing is not guaranteed. In fact, the order is guaranteed to be wrong if more than one message is processed at the same time; when a new message arrives before an earlier one has been completely processed then the new message will take priority, the first message on the stack will be the last one processed.

Apart from the difficulties of processing the messages in the wrong order, using this framework for lengthy conversations between the same agents, or a large number of agents requesting information at the same time, could lead to heavy use of the stack and heap, possibly culminating in a memory allocation failure. This made this protocol unsuitable for more sophisticated agent applications, although it worked well in this simple system.

A further problem was the occasional lack of response to user input. Key and mouse button presses would sometimes not be processed or an untenable delay would occur, causing the user to repeat an action only for both key presses to be responded to. The message time-out period could be adjusted to minimise this effect but if the time-out was too short then message delivery could fail.

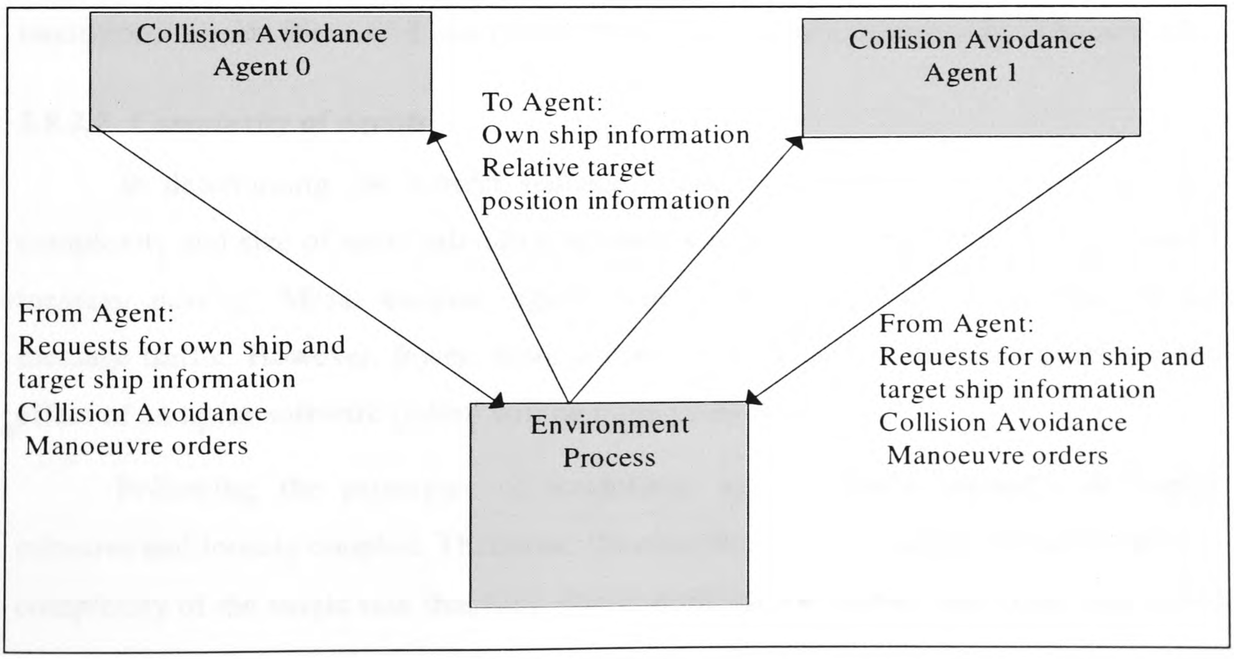


Figure 5-8 The information layer architecture V1.0

5.9.2 The challenges for a better message framework

As more agents were introduced it became apparent that a more robust message passing system was needed. Delivery of some messages began to fail due to collisions, time outs and memory allocation failures.

The accuracy and speed at which the messages are delivered in a continuous simulation is of utmost importance (Kuroda et al. 1994). Fault tolerance is also an important aspect, particularly if a MAS stresses the platform it is running upon by using all the available processing power or allocating all the available memory. In this case the message system must be able to re-transmit the important messages, if a message is not delivered and the agents themselves must be able to cope with some loss of less crucial information.

5.9.2.1 Number of agents

The two agents in V1.0 were supplemented by an additional agent on each ship to perform track keeping. The collision avoidance agent and the track keeping agent co-operate, deviating from the track to perform avoidance manoeuvres. Agents were then added to stress the system and the complexity of the conversations increased. A major challenge is to develop a message framework that is able to cope with a large number of loquacious[1] agents (Staniford and Paton 1994) on a limited communication bandwidth.

5.9.2.2 Complexity of agents

In determining the correct balance between the number of agents and the complexity and size of each individual agent it is necessary to consider the overhead of message passing. More, simpler, agents will normally produce an increase in the message traffic. However, fewer, more complex, agents will reduce flexibility and the effect of an agent software failure will be more pronounced.

Following the principles of modularity agents should normally be highly cohesive and loosely coupled. Therefore, the complexity of the agents depends upon the complexity of the single task that they should perform. Of course, one major task could

[1] talkative agents, that communicate perhaps more than is really required.

then be sub-divided so that a number of sub-agents perform a discrete part of the complete task.

5.9.2.3 Frequency of messages

As the number of agents and hence the message traffic increased in v1.0 it became apparent that one or two messages were not being received.

However, the increase in message traffic relative to the number of agents is not a linear relationship. Adding one more ship, and two more agents increases the workload of each existing agent as well as adding new agents. Hence, to accommodate twenty agents a message framework has to be able to carry not five times as many messages as one to cope with four agents, but fifteen times as many or more; the actual value will depend upon how many ships are in view of one another.

Designing a message framework with this increased capacity, while the underlying bandwidth remains the same is an important consideration.

5.9.2.4 Message response

When an instructor is using an interactive instructor station to control a continuous simulation, accurate and timely response to the instructor's input is essential. In particular, it must be immediately apparent that, for example, a new course or engine telegraph setting has been successfully entered.

However, in a marine simulation, ships respond slowly to their controls and it is possible that a delay of one or two seconds in actually processing the manoeuvring commands will not be noticed. More important is that the changes must either be effected or a warning given.

In a different type of simulation, such as a flight or car simulator, this is not the case. Whether message passing agents can be successful in controlling a car in a continuous simulation will, among other things, rely upon sufficient bandwidth to obtain rapid message response.

- Response Time

Ideally, the response time should be deterministic with response guaranteed within a specified time, at least for important messages. This is currently not the case and it is difficult to see how an agent system can provide this deterministic response. Other considerations such as the systems ability to recover from agent software failures mean that agent processes should run in a protected Operating System mode, this gives other processes priority. Additionally, if the messages are to be queued deterministic response will not be possible. This is one reason why critical messages should not be queued.

Probably, the best that can be hoped for is that the message passing system has priority over most other processing for that agent. If a message arrives it should be immediately processed or stored and the successful reception acknowledged. If the message requires an answer then, if possible, this should be given before any further planning is done.

- Response Accuracy

Equally important to response time is response accuracy (Kuroda et al. 1994). If a message is corrupted then it should not be delivered. The failure should be flagged and the correct data re-transmitted.

5.9.2.5 Message efficiency

The message passing protocol has to deal with a large number of messages passing between the applications. As several agents are able to request transmission almost simultaneously it is essential that the messages are passed quickly, keeping the 'busy' time low and collisions to a minimum.

- Encoding/Decoding

The encoding and decoding are designed to minimise the time that transmission is busy; very little actual processing is performed inside transmit/receive procedures. The data is collected and encoded into a message block. This is then copied into a DDE data block and then transmitted via DDE.

Upon arrival the messages are checked for validity. If very little processing is required or the message is urgent then it is dealt with immediately. Otherwise it is stored for processing later. During this time the link returns busy and no other messages can be received.

- Transmission

If the message is valid and there are no transmission warnings then a flag denoting successful delivery is returned to the sending agent. This permits the sending agent to re-transmit if an error occurs. The re-transmission time is arrived at by adding a random delay to the time at which the failure occurs. This appears to work for the number of messages currently transmitted, although a more sophisticated delay scheme, which is designed to minimise re-transmission collisions, might be needed in a more general MAS. Tuning the frequency and number of retries to the system performance and importance of a message could also improve efficiency.

5.9.2.6 Fault tolerance in message passing

The ability to cope with message collisions, unexpected termination of conversations, and a message system overload (where too many processes attempt to transmit simultaneously) are desirable features for a MAS. Some other systems have highlighted this requirement (Khedro and Genesereth 1994; Lee et al. 1993). In addition, a MAS may be able to perform automatic reconfiguration recovery (Levy and Rosenschein 1991). This is where a system is able to recover from broken links and lost nodes by sending messages via a different route. However, the usual requirements of a general inter-process message system such as DDE are less demanding. User driven requests on a single machine usually emanate from a single source. When retrieving data the process will then display a busy signal until the transaction is completed.

A local area network closely mirrors the activity in a MAS and many of the problems encountered have been solved in network protocols, although few truly deterministic networks have been created. In a continuous simulation the problem is exacerbated by the need for rapid response. As the complexity of the simulation rises the performance should degrade gracefully (Reddy and Moon 1995).

- Loss of one agent in the conversation

Ideally, if one process suffers a run-time error and terminates while engaged in a conversation, it should be possible for the other processes to detect the problem and continue running.

- Collisions

Messages that arrive while the server is already processing a previous message must be re-transmitted by the sender. Therefore, the decoding must be done as rapidly as possible in order to minimise retries.

- Re-transmission

If a message transmission fails then the sending application should retry after a random period. The actual number of attempts at re-transmission that should be made depends upon the importance of the message, the bandwidth, the message size, the message frequency, the time-out duration, and the number of agents.

5.9.2.7 Message Over-load

There may be several reasons for an apparent increase in message traffic within the MARINES MAS:

- the agents may actually be generating more messages;
- a virtual memory manager will generate more disk activity as more linear memory is used by applications, this slows the overall performance of the machine. Therefore, storing more and more messages may actually generate slower performance, more message failures and more memory allocation until the system locks altogether;
- one badly behaved process may hold the processor for longer than is normal, this may lead to a temporary increase in traffic as the agents try to send their backlog of messages. This may also occur if there is a software error in one agent process, the instructor may terminate the errant process and return processing to the other agents;
- more agents may be launched than the message system can support.

As can be seen the increase in message activity may be very short term. If this is the case then it is probable that the message delivery will catch up. However, if the number of messages being held for processing by an agent is growing over a longer period of time, the number of transmission failures is growing or the system is approaching memory allocation failure then some strategy should be in place to ensure graceful performance degradation.

5.9.2.8 Adapting to incomplete information

The loss of some messages makes it essential that the agents are able to make attempts at planning with incomplete information. This also implies a prioritised message system and/or an agent that is able to re-request critical information.

5.9.3 The refined MARINES message framework

In MARINES version 1.07 the number of agents was increased from four to twenty, two on each of ten ships. Under Windows 3.1, on a Pentium 75, I consider this to be very close to the upper limit that can usefully operate, in real time, at this level of complexity. Beyond this point the system becomes too sluggish to be used effectively.

Three major changes were necessary to cope with the increased message traffic, due to the introduction of twenty agents. Namely, asynchronous transmissions, the use of a FIFO[2] message queue and the re-transmission of failed messages. The use of asynchronous DDE messages speeds up the communications. However, replies have to be generated and returned separately, they cannot be given immediately. The FIFO message queue is used to store messages that are not urgent, ensuring the replies are sent out in order. A busy flag prevents new messages being delivered while a freshly arrived message is moved to the queue, or an urgent message is being processed. This may result in a message not being delivered. An additional communication layer was added to the message framework, enabling the sending agent to determine information about the message transmission and choose whether to re-transmit or discard a message.

5.9.3.1 The protocols

Version 1.07 of MARINES uses a mixture of remote invocation and asynchronous message passing (Burns and Davies 1993), depending upon the type of message being delivered. Asynchronous transmissions permit the sending agent to

[2] First In First Out queues where the messages are processed in the order they arrived in.

continue processing while confirmation messages are awaited, this improves the response to user interaction and minimises the time that the receiver is busy. Therefore, less message collisions occur. Remote invocation is used only for urgent messages or those that can be processed quickly; messages that require more than minimal processing are placed in a message queue for processing at a later time.

5.9.3.2 The transmission layer

As in version 1.0 each agent process is connected as a client to the central MARINES environment and the Environment then connects back to the agent server. However, in version 1.07 the agents may also be connected to each other if desired as shown in Figure 5-9.

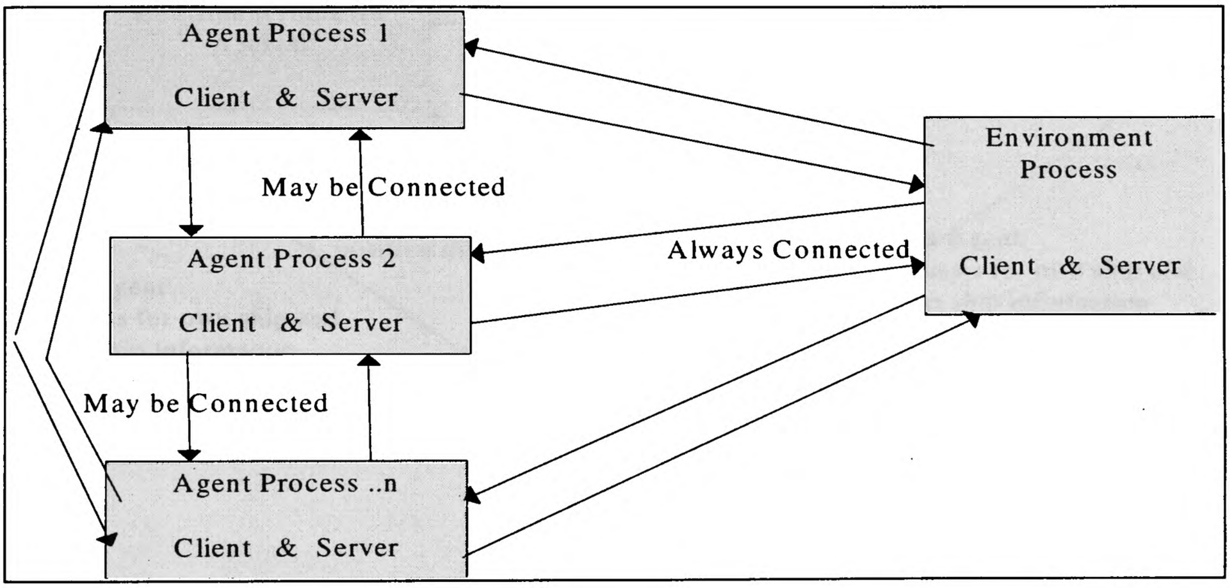


Figure 5-9 The transmission layer architecture V1.07

5.9.3.3 MARINES message protocol, the communication layer

In version 1.07, in addition to the content of the message to be passed between agents, each message block contains information about the conversation: the conversation identifier; the ID of the sending and receiving processes; the number of times that transmission has been attempted; and the last time that an attempt at sending the message was made.

This information is used by a MARINES agent in order to ensure that important messages are delivered. Thus, an important message might be timed out by the transmission layer itself and then re-transmitted by the agent, while a less important message might not be re-transmitted. This has been found to considerably improve system response over blanket re-transmission, while ensuring that essential information is not discarded.

5.9.3.4 Message content , the information layer

Several new messages have been added to the information layer, shown in Figure 5-10, to enable inter-agent communication and the agents have become more loquacious, putting a greater strain upon the transmission and communications layers.

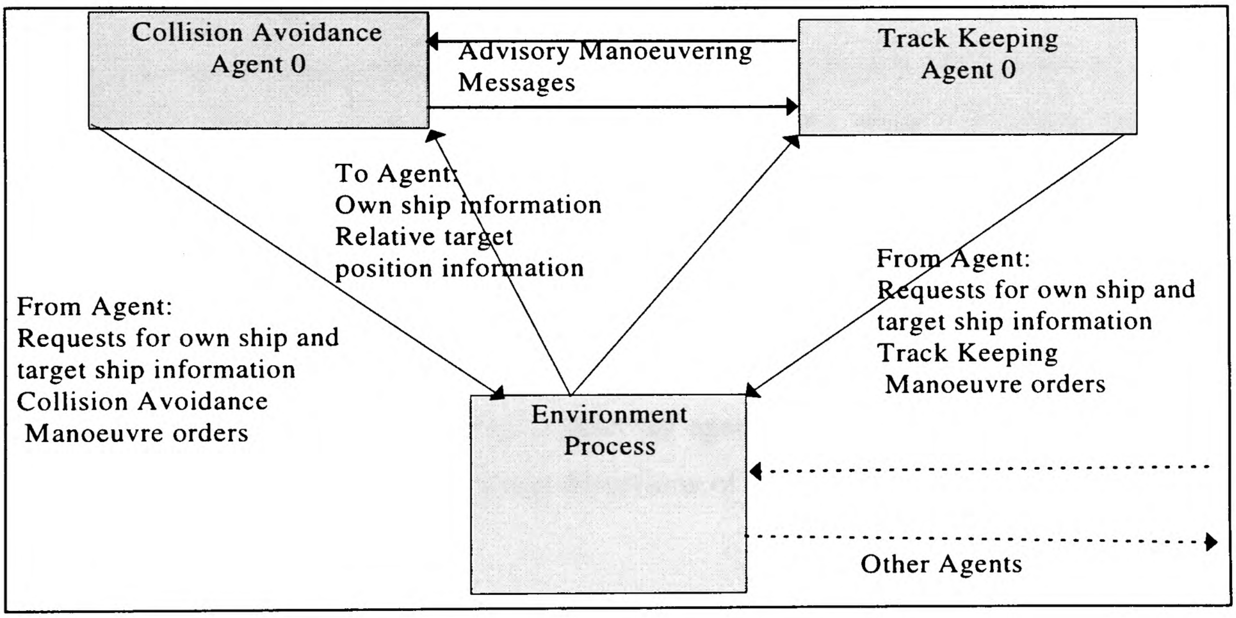


Figure 5-10 The information layer architecture V1.07

5.9.3.5 Frequency of messages

The collision avoidance agent sends the environment messages requesting details of any visible ships and information about the ship that the agent was controlling. This was found to operate best with information at approximately six second intervals; this permits the agents to ascertain changes to a target ship's course or speed reasonably quickly, without constantly misinterpreting minor changes due to the yaw or small data errors. It may be necessary to modify this interval according to the detection range, number of targets and proximity to fixed dangers.

Messages are also sent to retrieve information to update the ships' instruments that are displayed in the agent interface and messages to assist with instructor monitoring. Because of their interactive nature these messages were found to be needed more frequently, at least once every three seconds.

5.9.3.6 Automatic Reconfiguration

In MARINES v1.07 if an instance of a collision avoidance agent fails then the track keeping agent will detect this and send manoeuvring messages directly to the instructor environment. This minimises the damage caused by the loss of an agent.

5.9.3.7 Adapting to incomplete information

In a worst case scenario the collision avoidance agents in MARINES must make a reactive manoeuvre to avoid a ship that is detected late. This is a simple avoidance manoeuvre if a target ship infringes the domain (Goodwin 1975) around the agent's vessel.

A more normal scenario is that a ship may be detected, the information stored, and then a subsequent scan may fail to detect the target. The agent will then keep the previous information until the ship is detected again and then perform the calculations for the time period between the actual detections of the target.

5.9.3.8 Re-transmission

After a limited amount of testing it has been found that for the Version 1.07 implementation, the re-transmission should only be attempted three times. Trying to re-transmit more than this actually results in more collisions and a greater number of message failures. Beyond this, an agent can try sending another request if essential information is not delivered.

DDE appears to provide an accurate message passing mechanism; no data corruption has been detected. Therefore, no additional information has been added to messages to allow for accurate delivery. However, it is important that the messages are validated on arrival, at least to check that values are legal for the program, this reduces the opportunity for errors in one process producing errors in another. For example, a ships telegraph setting should be between -100%..+100% and this is checked on arrival of the message.

5.10 MARINES and other MAS test beds and frameworks

In putting the MARINES framework in context with other MAS systems the following points should be borne in mind: many MAS test beds are actually implemented as a single process; many of the test beds are not subjected to the challenge of imperfect message passing; many of the test beds perform all the operations synchronously, messages are updated at a specific point in the cycle and are based upon the same information for all the processes; many theoretical frameworks are actually intended for a fully distributed system on separate processors; only a few frameworks have been successfully used in fully working systems; some frameworks are purely theoretical.

Little information is available about the message frameworks employed in single process, single thread MASs. Even if this was widely available many of the challenges of a multi-process system would not be considered, lost messages, deadlocks, livelocks, etc.

Blackboard systems have been used for MAS implementations. However, the blackboard model is one of solving one large problem by a large number of medium grained agents. In MARINES the decision was to follow the coarser grained model similar to that used by Fergusson (1994) and the InteRRaP model (Muller et al. 1994). In any case, global blackboards usually only hold common results, while private blackboards or scratch pads hold the temporary values. This does not really match with the problem of simulating marine collision avoidance performed by the individual officers of the watch on separate ships and thus a Blackboard system was not used.

In MARINES the decision was taken to use a message passing architecture for a number of reasons. Firstly, a truly distributed system on separate computers would be unable to have a shared memory mechanism without some message system to obtain the data. Secondly, developments in Microsoft Windows make it more and more difficult to create memory that is shared between processes, however, it does provide standard message passing protocols such as DDE. Thirdly, protection can be offered to the data that is encapsulated within an object. Fourth, Hewitt and Leiberman (1984) and Maitre and Laasri (1990) suggest that message passing is more suitable than shared data, even for a Blackboard system. Finally, message passing has been successfully used in Multi-Agent Systems such as ARCHON (O'Hare and Jennings 1996) and APRIL (McCabe and Clark 1994). The major disadvantage of message passing is perceived as the performance overhead of the message protocol.

In MARINES each agent is a DDE server and a DDE client. Therefore, although the topology shown in Figure 5-10 is currently used to connect the agents, any configuration can be supported as shown in Figure 5-9. However, the information layer messages would need to be adapted to accommodate the desired communication.

5.10.1 MARINES Hybrid Architecture

In chapter 3 both an Object-Centred Autonomous Architecture (Steeb et al., 1981) and a Hierarchical Architecture were described. The MARINES message framework is a hybrid of the two. Each ship sub-system is object centred and autonomous. There is no direct inter-ship communication, although the individual agents infer the intentions of the other ships. In most cases the agents are inherently co-operative, the collision regulations being designed to assist ships in avoiding each other. This also follows Steeb's recommendation for a complete set of rules of the road to assist the modelling. Within each ship sub-system the agents have a hierarchical structure. The track keeping agent advises the collision avoidance agent where to steer to maintain track. The high communications overhead described by Steeb is overcome in two ways. Firstly, the number of agents that communicate with each other is restricted to the agent groups on each ship; in the case of MARINES v1.07 each group consists of two agents. Secondly, the communication is restricted to a minimal set; the set is based upon the normal co-operation between bridge teams.

5.11 Summary

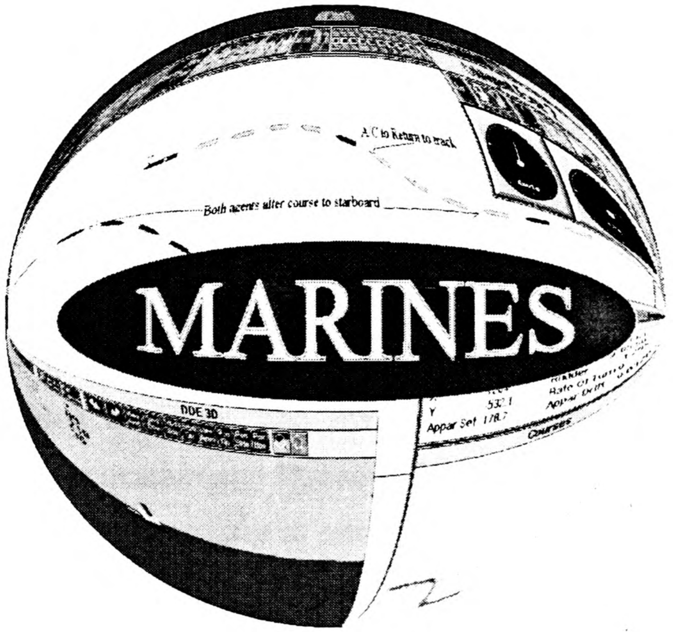
This chapter has described the MARINES system, firstly in a conceptual manner and then an overview of the implementation has been given. Each software process has *been briefly described, including the instructor environment, the 3D view and the agents. These are considered in more detail in chapters 6 and 7.*

After this the selection of the main techniques and tools used in the development of MARINES have been described: Object Orientation; Message Passing; Microsoft Windows; Dynamic Data Exchange; and C++. Microsoft Windows 3.1 was chosen to provide a co-operative multi-tasking operating environment, availability and familiarity of the author influencing the decision. The need for efficient code and the use of many similar objects, such as ships, influenced the selection of C++. The chosen tools and techniques are intended to provide a flexible Multi-Agent System with agents as discrete processes.

Following this, the message passing framework has been described in detail. V1.0 of the framework became unstable as the number of messages increased. This was found to be due to the immediate processing of incoming messages. The relatively long processing time caused time-outs to occur in the DDE and considerable use of the stack as messages arrived. In Version 1.07 asynchronous transmissions and a FIFO message queue were introduced. This permitted newly arrived messages to be stored until older messages were processed. There is, however, a mechanism for dealing with high priority messages immediately. Messages are also re-transmitted if the original transmission fails. This is limited to three attempts for the majority of messages to prevent overloading the framework. After this, it is the responsibility of the agent to request essential information a second or subsequent time.

The similarities and differences between MARINES and other MAS frameworks have been highlighted. The MARINES framework is essentially a hybrid of an object centred autonomous architecture and a hierarchical architecture, this has been described in more detail in chapter 3.

Chapter Six



MARINES Instructor Environment

6. MARINES Instructor Environment

6.1 Overview

This chapter gives a high level description of the MARINES instructor environment process. Both a conceptual view and an object model are shown and the similarity between the two is briefly discussed. The internal architecture of the environment including the object oriented mathematical model is shown.

After this, in section 6.3.9, the instructor environment interface is described. The reasoning behind the design and the use of each control is briefly considered.

6.2 Introduction

The instructor environment provides a subset of the capabilities found in the instructor station of a bridge simulator. This includes, computer generated target ships, a plan view of the scenario, the ability to alter environmental conditions, run and pause the simulation and display the tracks of the ships. Additionally, the ability to run the simulation at speeds up to five times faster than world time has been incorporated to speed up the evaluation exercises. The environment process runs under Microsoft Windows as a 16 bit application and Dynamic Data Exchange connectivity has been provided. This allows other processes to retrieve information about their environment and control the computer generated target ships. The program has been designed using Rumbaugh's Object Modelling Technique and is written in C++.

6.3 Architecture

Figure 6-1 shows a context diagram of the MARINES instructor environment. The instructor issues commands, setting the set and drift of the current, starting the exercises etc. A plan view of the scenario is displayed upon the screen, together with the controls available to the instructor. Intelligent agents can connect to the environment, retrieving environmental information and setting the controls of the ship they are controlling.

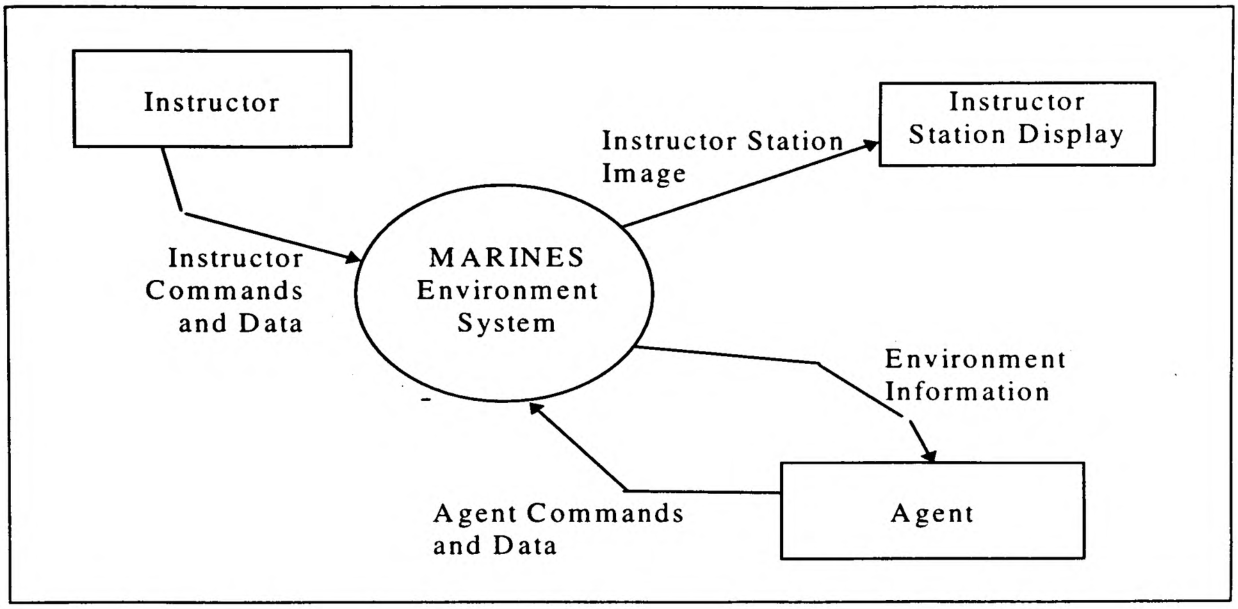


Figure 6-1 Context diagram of the MARINES Instructor environment.

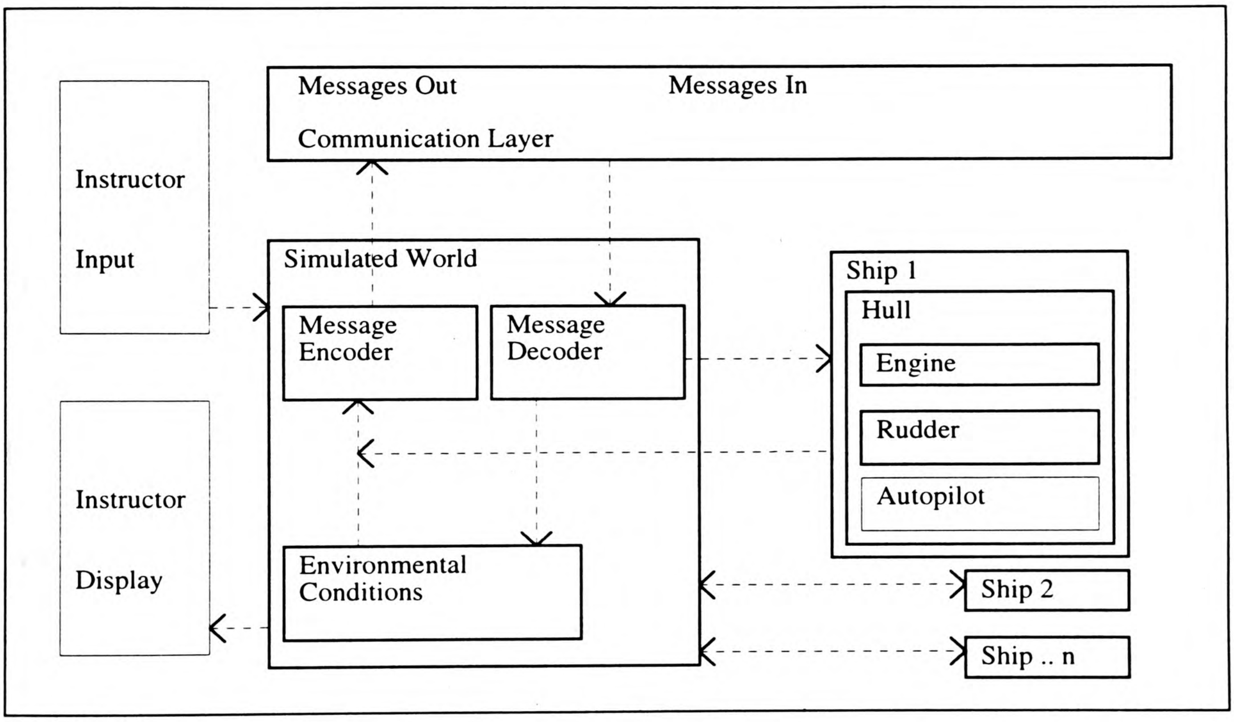


Figure 6-2 Conceptual view of the Instructor Station Environment Architecture

The conceptual view, in Figure 6-2, shows an abstraction of the instructor station environment architecture. This provides a general overview, it does not show the physical structure of the code components. A high level view of the actual components and their interconnection is given in Figure 6-3 and Figure 6-4.

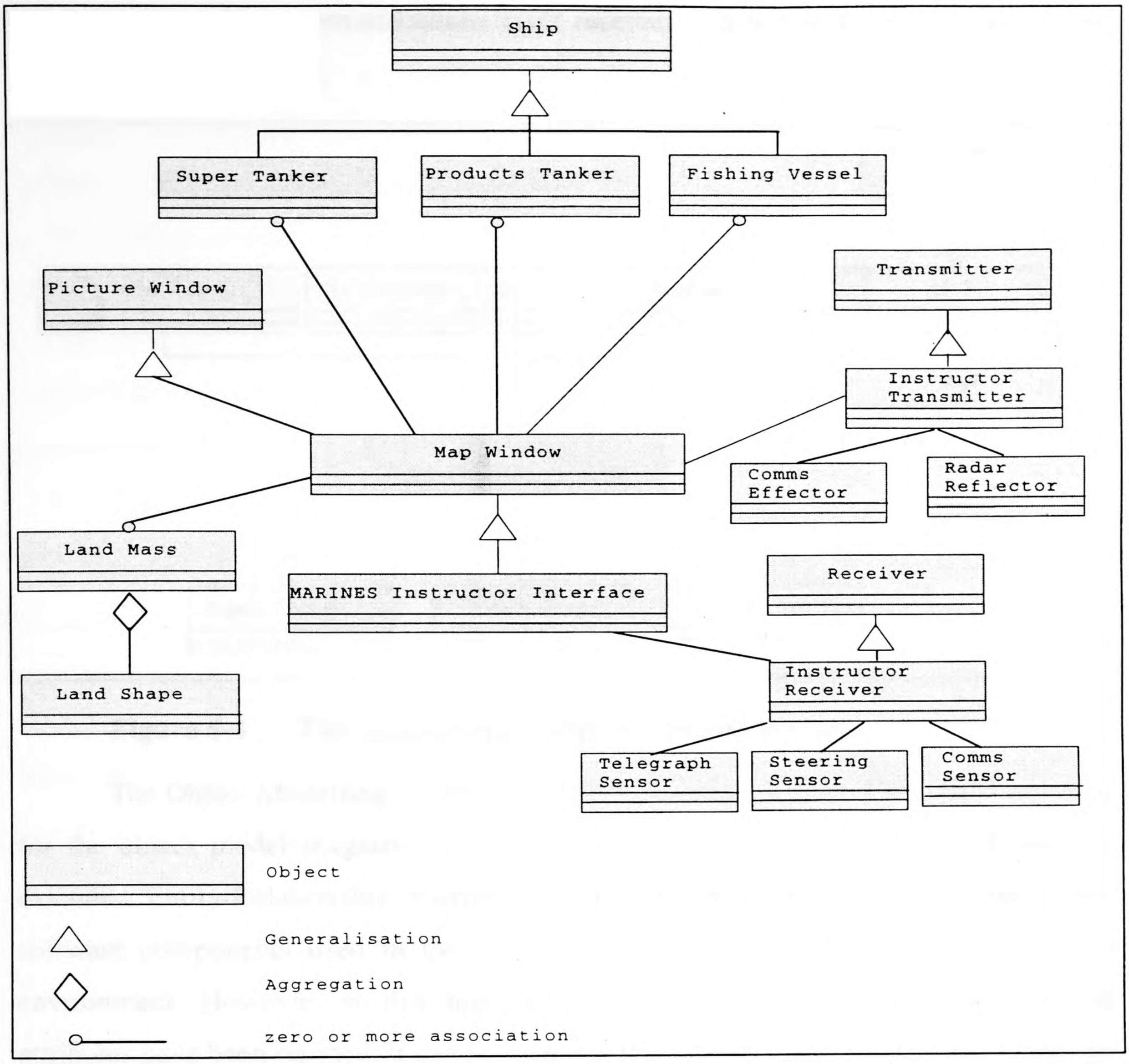


Figure 6-3 An Object Model of the Architecture for the MARINES Instructor Station Environment

The environment, or simulated world, that the ships inhabit contains land and sea areas. Unlike the majority of MAS this is not a grid based environment. The simulated area has no distinct boundaries, except those formed by the computing limits of the variables used in the software construction. At present the ships are not constrained by their draught, they are able to travel anywhere in the area. The current affecting the ships can be adjusted to affect the motion of the ships. The ships themselves calculate their own movement around the area for very small periods of elapsed simulation time. The actual 'dt' between steps depends upon a large number of factors such as, simulation rate, the number of ships moving and other processes using processor time. The communications layer receives and transmits information to and from the agent processes.

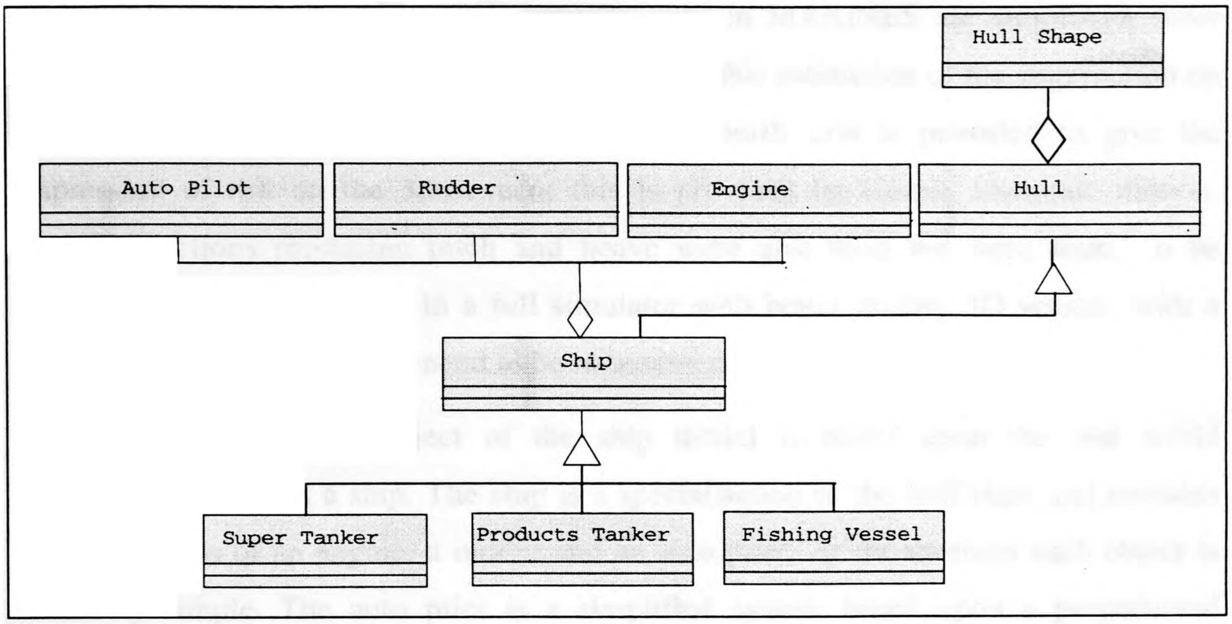


Figure 6-4 The mathematical ship model architecture

The Object Modelling Technique (OMT) (Rumbaugh et al. 1991) has been used for the object model diagrams shown in Figure 6-3 and Figure 6-4, OMT uses an extended Entity-Relationship diagram syntax. This is a model showing the major software components used in the construction of the MARINES instructor station environment. However, at this high level of design, details of the methods and attributes have been omitted. It can be seen that this Object Oriented approach to design produces very similar components to those in the conceptual view, this makes the top level analysis and modelling quite straight forward when compared to functional decomposition of the problem. This is especially so when the software is to run in an event driven multi-tasking environment, and is developed in an application framework such as Borland's Object Windows Library (OWL). Message driven events and response functions in Windows are difficult to model in a standard data flow design. Some data flow techniques with extended syntax do attempt to model control (Goldsmith 1993), these were not considered at the time of this design.

The components making up the environment will now be discussed individually.

6.3.1 Mathematical Ship Models

The mathematical ship models have been designed and built using an Object Oriented Architecture. The models actually used in MARINES are simplified, three degrees of freedom are provided to obtain a reasonable estimation of the ship motion on a 2D display. A very simple motion about a fourth axis is provided to give the impression of roll on the 3D screen; this is provided by simple harmonic motion. Simple functions producing pitch and heave were also tried but were found to be unnecessary for MARINES; in a full simulator with better quality 3D visuals, with a higher frame rate, this would need to be re-assessed.

Each component object of the ship model is based upon the real world components found in a ship. The ship is a specialisation of the hull class and contains an aggregation of an engine, a rudder and an auto pilot. At the moment each object is relatively simple. The auto pilot is a simplified system based upon a proportional integral derivative (PID) controller. An error signal is obtained by the difference between the desired course and the actual course and this is used to determine how much helm to apply and the rate of turn is used to determine the counter helm. An integral term is not used for the weather helm at present; note, however, that a true integrator is seldom used in modern auto pilots used aboard real ships. The engine component is based upon a power curve and the revs arrived at by a simple delay loop. The rudder movement also relies upon a simple delay, the rudder area is a constant for the ship type. The hull contains constants for length, beam and frictional area etc., the hull shape contains a number of polygons representing a plan view of the basic hull and accommodation. Figure 6-4 shows the actual architecture used in the design of the software, this is a simplified abstraction of the a real world ship.

More importantly, it is possible to replace the components with more sophisticated models, if this is desired. Careful interface design is essential to allow this to be done. Using Object Oriented analysis, an abstraction of the real-world interface of the components produces a basic interface. Therefore, the engine interface allows for setting the throttle and determining the throttle setting, revs and power output. The rudder interface allows the helm angle to be set and the present rudder angle and effect retrieved. The hull provides the friction and shape. The aggregation of components into the ship has an interface that allows the ship to be controlled as a normal ship would; the helm and the telegraph provide the main interface, environmental conditions are also passed. Instruments need to display speed, position, rate of turn, engine revs, etc., therefore, these values are obtainable via this interface. This interface is considered adequate for the computer generated target ships in MARINES version 1.07. In the future, as simulations become more sophisticated, communications bandwidth increases, or for simulations requiring even more accurate target ship motion there may be the need for a more complex interface. One example that could be considered in the future is of an agent, or instructor, that will monitor the simulation and introduce faults to the target ships, as the instructor does now to the student ship; engine or steering failure at an in-opportune moment, etc. However, for the moment, computer performance has to be considered and, it is suggested, that this only be done if it is necessary, especially, if a large number of computer generated target ships are required.

In version 1.07 of MARINES the ship types all use the same basic components. The only differences are the default manoeuvring constants and the polygons for the hull shapes; these give a generic model for each class of ship. The ship class itself contains the functionality to calculate the motion of the ship based upon the output from these components. The manoeuvring characteristics of the ship models are shown in chapter eight and discussed further in chapter nine.

6.3.2 Land Mass

Each land mass is made up of one or more land shapes. At present, these include land shapes for the beaches and the higher areas. These are displayed in the map window in much the same way as the ship shapes.

6.3.2.1 Land Shapes

In the MARINES v1.07 each land shape is actually a polygon. The polygon class is used for the ship shapes, land shapes and the needles on the gauges in the collision avoidance agent display.

6.3.3 Picture Window

The picture window provides a standardised drawing area. The area is an off-screen buffer the contents of which are then copied (bitblitted) to the screen, reducing the discernible flickering when constant motion is occurring. In fact this is implemented as a device dependent bitmap under Microsoft Windows. The object provides a device context and a number of pens and brushes. Additional functionality is provided for scaling and clipping polygons before they are needed. Windows will perform most clipping but this is done at the time of drawing and can be slow for filled figures. It is also better to clip to the screen before converting the numbers from floating point to integer maths to prevent overflow; otherwise large values may be out of the integer range and therefore become corrupted. Finally, the picture window also standardises the map mode and bitblit functions inverting and scaling the co-ordinate system to start at the bottom left corner.

6.3.4 Map window

The map window inherits from the picture window and permits the land and ships to be drawn. However, in Microsoft Windows, the windows are each provided with a message queue. Therefore, a window normally acts as the central hub for administering input and output information; event handling functions are also attached to the window object. In this way, the map window is more than just a simple display mechanism. It also deals with requests from the receiver's sensors for information about the environment. Commands from the agents to set the agent's ship telegraph or desired course are decoded and passed to the relevant ship for processing via the map window. If response is required then the map window also provides functions to encode the environment messages and request transmission.

6.3.5 The transmitter

General connection, disconnection and transmission protocols are provided by the transmitter. In MARINES v1.07 the basic transmitter is a DDEML client implementation that permits, connection, communication and disconnection with a number of servers. A linked list holds the details of all the servers that the transmitter is connected to. All the messages in v1.07 are sent asynchronously, this speeds up the communication as no return data is expected.

6.3.6 The instructor transmitter

The instructor transmitter inherits the transmission protocols from the basic transmitter and adds functionality for the specific messages that the instructor station transmits to agents and the 3D_DDE process. The messages simulate various generic effectors, although additional filters can be implemented on reception. For example, the radar message will contain the range and bearing of a target within a certain range. Both the range and bearing will be accurate. Upon reception a filter in the sensor may reduce the accuracy or apply errors pertaining to a specific radar. This feature has not been used in v1.07 of MARINES, other features taking precedence in this research.

6.3.7 The receiver

The receiver is a DDE server implementation. A list of connected clients is maintained. Clients can connect and disconnect from the server and Poke messages via this server callback to the application. DDE advise loops are not supported due to the overhead of sending, possibly unneeded, data very frequently.

6.3.8 The instructor receiver

The instructor receiver adds the necessary functionality to deal with requests for data and commands for the ships. The messages are checked upon receipt to ensure that they contain legal data; this reduces the danger of errors in one process being transmitted to another.

6.3.9 The instructor interface

The instructor environment interface, shown in Figure 6-5, consists of a plan view of the scenario and a number of control buttons. The plan view shows the positions of the ships, land and sea, allowing the instructor to tell at a glance the relative positions and headings of the ships. The control buttons are grouped according to function, allowing the instructor to control and monitor the exercise as desired. In this experimental system no confirmation is sought for irreversible actions. For example, if 'Exit' is chosen the system will simply close down. Relatively straight forward changes would need to be made to correct this in a commercial system. Context sensitive help and flyover hints would also be desirable. However, some configuration would be beneficial to prevent experienced users being overloaded with unnecessary confirmation messages.

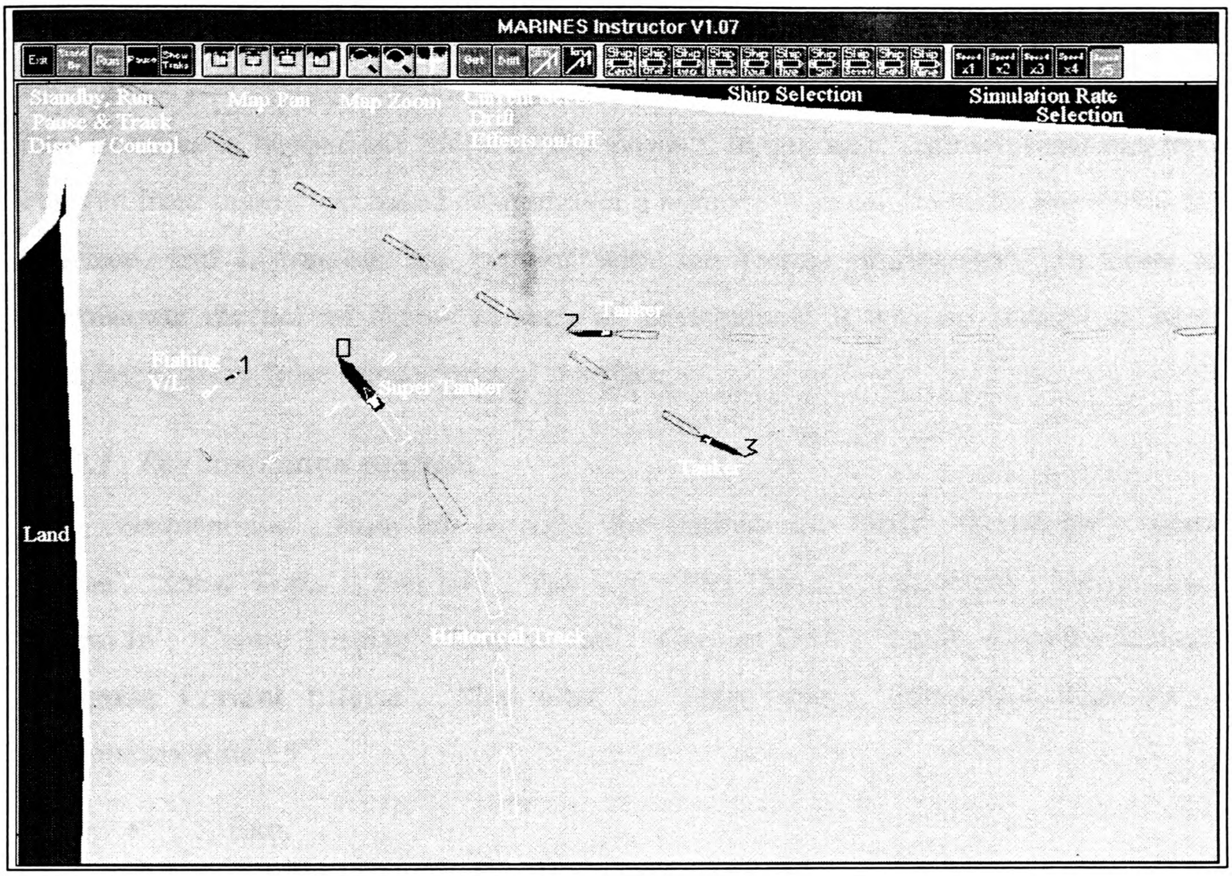


Figure 6-5 The MARINES instructor environment interface.

The MARINES instructor environment interface is somewhat similar to the interface used in the Mardyn Ship Handling and Port Design simulator. However, the functionality is a sub-set of that found in a complete simulator and, unlike the MS-DOS program used in the Mardyn system, this program runs under Microsoft Windows. Additionally, some of the functions driven by menus in the Mardyn system are provided by WIMP style buttons and icons. Shneiderman(1987) shows that users make a high percentage of errors performing relatively simple tasks; even experienced users made mistakes in 10% of a simple text processing task, with only 15 commands. Schneiderman also shows that users made less errors using a GDI Window style interface. Neilsen(1993) states that most "*graphical user interfaces have better* *usability characteristics in general than character-based interfaces, especially with respect to learnability for novice users".* Therefore, this WIMP environment was considered desirable. The size of the title bar, tool bar, buttons, etc., are all the standard size used by Microsoft applications, maintaining a consistent look and feel.

This provides the instructor environment with an interface that retains the appearance and usability of a conventional instructor station. Design of a completely new interface is beyond the scope of this project. In any case, instructor stations have evolved from purely text based designs over a number of years, gradually improving the interface, and instructors are familiar with the format. Furthermore, in order to demonstrate the use of agents in such an environment it was not wished to move completely away from a conventional interface.

6.3.9.1 The simulation controls

Sequentially , from left to right, the buttons are: 'Exit'; 'Stand By'; 'Run'; 'Pause', 'Show Traks'; 'Pan left'; 'Pan Up'; 'Pan Down'; 'Pan Right'; 'Zoom Out'; 'Zoom In'; 'Centre Display'; 'Current Set'; 'Current Drift'; 'Enable Current Effects'; 'Suppress Current Effects'; 'Ship One' .. 'Ship Nine'; 'Simulation Rate *1' .. 'Simulation Rate *5'.

- Exit

The Exit control is used to close the simulation. In a simulator a check must be made that this is what is required. However, this is not done in this research version so care must be taken that this is the desired course of action. Additionally, the agents are not closed down by this control in version 1.07.

- Standby, Run, Pause and Show Tracks

In standby mode changes to ship speed and course take immediate effect and the ships do not move; this permits the instructor to set the exercise up. Once run is pressed then the changes take a normal duration to take effect and the ships move normally. The run can be paused using the Pause button, the entire simulation stops, although the instructor can still implement some changes. In the paused mode the historical tracks of the ships can be displayed using the Show Tracks button. The positions are displayed at intervals of one minute (Simulation time).

- **Pan, Zoom and Centre display**

The Pan buttons permit the plan view display to be moved left, right, up and down in steps. The Zoom buttons increase and decrease the scale of the plan view. The Centre Display button positions the view at the centre of the map.

- **Set, Drift and Effects of Current On/Off**

Pressing the Set button displays a data entry dialog box that permits a new current direction to be entered. Similarly, the Drift button permits a new current rate to be entered via a dialog box. The same current affects the entire data base. However, for research purposes, the effects of the current can be individually suppressed for each specific ship. Pressing the Current Off button suppresses the current for the selected ship.

- **Ship Selection controls**

Pressing the relevant ship button centres the display at the selected ship. A message is also sent to the relevant agents so that they can make themselves visible. The agents for the previously selected ship will hide themselves when they determine they are no longer selected.

- **Simulation Rate Controls**

In order to speed up research exercises it is possible to run the simulation at a rate of up to five times world time. However, this feature must be used with care. If ten ships and twenty agents are run at fast time the communications overhead is considerable. The effect is exacerbated by the more frequent storage of track information. On a Pentium 75 the loss of messages at a rate of five times world time can severely affect the results. It has been found that four agents can safely be connected, to two ships, at five times world time and twenty agents, to ten ships, at world time on a Pentium 75.

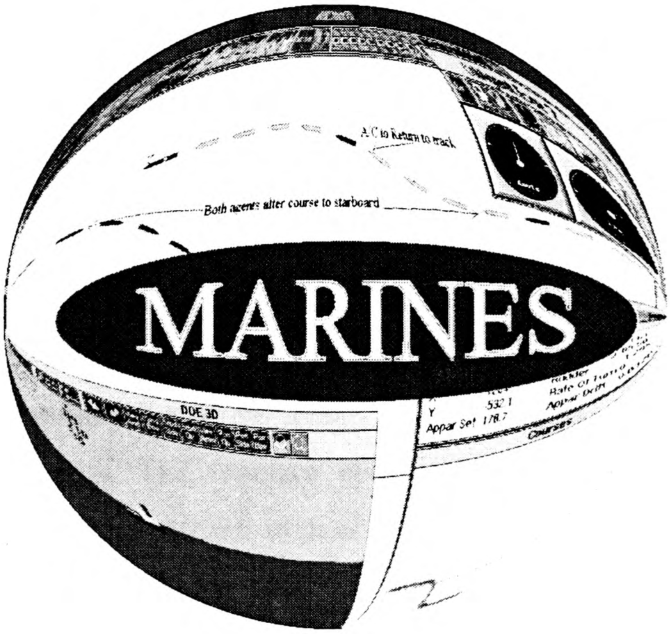
6.4 Summary

This chapter has described the high level design of the MARINES instructor station. The station supplies a sub-set of the features found in a real marine simulator instructor station and some additional functions for research purposes. It also forms the dynamically changing environment that the agents connect to.

It can be seen that there is a close correlation between the conceptual view of the architecture and the Object Oriented design of the architecture. Functional decomposition would have separated the design into less coherent units, making this similarity less apparent. The use of Object Oriented design has also been useful in facilitating multiple instances of the different ship models. Ten instances of ships are used in MARINES v1.07, based upon three different models. Presently, generic models of a super tanker, a tanker and a fishing vessel have been created.

The interface to the MARINES instructor environment has been explained. Essentially, it permits control of the exercises and manipulation of the plan view. Additionally, it allows the experimenter to adjust the direction and rate of the current and enable and disable the effects of current upon the ships.

Chapter Seven



The Collision Avoidance and Track Keeping Agents

7. The Collision Avoidance and Track Keeping Agents

7.1 Overview

There are two basic agent types in MARINES, the first built is a collision avoidance[1] agent and the second performs track keeping[2]. To be effective the agents have to collaborate to keep the ship on track and perform manoeuvres, when necessary, to avoid other ships.

This chapter discusses the hierarchy between the agents and then the internal architecture of each of the two classes of agent. The individual components are described and then the interface between the agent and the instructor is considered. Many of the components are derived from classes used in the MARINES instructor station described in the last chapter, therefore the descriptions are kept brief. The designs of the architecture are then discussed in the context of the agent literature found in chapter 2.

7.2 Introduction

The internal architecture of the agents in MARINES consists of both reactive and planning components. The reactive component takes care of exogenous and unplanned events (Hanks et al. 1993), such as an approaching ship that does not comply with the rules, or an unwarranted deviation from the desired track. The planning component extrapolates historical data to determine when and how to make manoeuvres. Therefore, a store of historical data and inferred facts is maintained. A deliberation engine determines whether the reactive or planning component is called according to the incoming information. The other essential components are a transmitter and a receiver for the messages that are passed to the environment and the other agents. These communications systems process messages through simulated sensors and effectors to provide a subset of the information that would be available to the navigator of a ship. The interface between the agent and the instructor is based upon

[1] A simple description of the marine collision avoidance problem is given in appendix A

[2] The track keeping problem is also outlined briefly in appendix B

the type of information that a navigator would have, such as course, speed, rate of turn, etc.

In order to create these agents as Microsoft Windows processes, a Window forms the central focus of the interface and provides response functions for button press events, etc. This Window inherits the properties of the agent to create a Windows based agent class. The decision to build the agent separately from the Window was to reduce coupling and increase cohesion, reducing the dependence upon a single operating system. However, in practice, the inherited classes also had to be implemented as OWL (Object Windows Library) classes and the use of DDE closely ties much of this program code to the Windows operating environment.

7.3 Starting and stopping agent processes

The MARINES system is a test environment, and the agents can be started and stopped at run time. Therefore, an Exit button occurs on each Agent process. This permits the experimenter great flexibility in configuring the system, and also facilitates rapid changes that the experimenter wishes to make. However, no warning messages are displayed if Exit is selected. This would be inappropriate in a real simulator as accidental termination might occur. In a commercial system, it would be preferable to provide a different mechanism for starting and stopping agents. One possible solution would be to have information about the set-up of the agents in an exercise configuration file and introduce levels of warnings and help messages that the instructor can configure to suit his/her experience.

7.4 Agent Hierarchy

In MARINES v1.07 there are two agents aboard each computer generated ship. One agent performs collision avoidance[3], the other performs track keeping[4]. The agents in MARINES follow normal practice for officers on a bridge watch; the collision avoidance agent takes command of the ship and the track keeping agent acts as an assistant, sending the collision avoidance agent advisory messages.

[3] A simple description of the marine collision avoidance problem is given in appendix A

[4] The track keeping problem is also outlined briefly in appendix B

This hierarchy provides a simple solution to conflict resolution between the agents. If the agents were allowed to be completely independent then when the collision avoidance agent alters course to miss an approaching vessel the navigation agent would immediately apply set to bring the ship back onto track. A more collaborative approach would be in keeping with agent literature but require a higher level of inter-agent communication and agent sophistication.

The agents are based upon the human navigators found on the bridge of a vessel, rather than on the ships themselves. The modelling is therefore based upon an analysis of the real world. It is the interaction of simplified real world factors that produces the complex simulation, rather than a heuristic model of the observed behaviour. The interaction of many relatively simple autonomous agents helps to produce a reasonably realistic, complex simulation. A heuristic model of the observed behaviour of the complete system would be more tightly coupled and possibly more difficult to achieve using conventional monolithic programming techniques.

7.5 Collision Avoidance Agent

Marine collision avoidance has been widely studied for a number of years by several research groups (Blackwell et al 1988; Blackwell and Stockel 1989, 1990; Blackwell et al 1991; James 1986; Smeaton and Coenen 1990). Among the most successful is the research of Grabowski's group (Grabowski 1990; Grabowski and Sanborn 1992, 1995a, 1995b; Sudehendar and Grabowski1996). A version of this system is fitted to Exxon Tankers operating in Valdiz sound, following the stranding of the Exxon Valdiz. This work has been described in chapter 2.

The MARINES project is focused on producing a realistic simulation, collision avoidance representing only a part of the work. Therefore, while MARINES contains similarities to the research of the other groups, the MARINES implementation is less sophisticated. Currently, automatic collision avoidance only applies for situations where two power driven vessels are approaching one another. Nevertheless, several of these situations may occur simultaneously and new rule bases for other ship types may be created.

The collision avoidance agent monitors approaching traffic and determines the danger that this represents using Smeaton's coefficient (Smeaton and Coenen 1990) as a guide as to when to take avoiding action. The actual amount to alter course is selected using a domain (Goodwin 1975) around the ship; the domain used is circular, reducing the computational complexity (Davis et al. 1980), this being considered adequate at present.

7.5.1 Architecture

Agent architecture usually follows either a vertical or horizontal scheme (Muller et al. 1994), as shown in chapter 3. Agents with a vertical architecture usually pass information to the reactive sub-system first. If this is not processed then it will be passed upwards to the planning sub-system. Agents with a horizontal architecture will process the information in parallel in both the reactive and planning sub-systems, a controlling process determining the action to take.

The MARINES collision avoidance agent, shown in Figure 7-1, is a hybrid of the two architectures. The deliberation engine determines whether immediate danger exists and passes the incoming information to either the reactive or the planning sub-system accordingly. Once again the design diagram shows the chosen architecture rather than precisely matching the real world it is abstracted from.

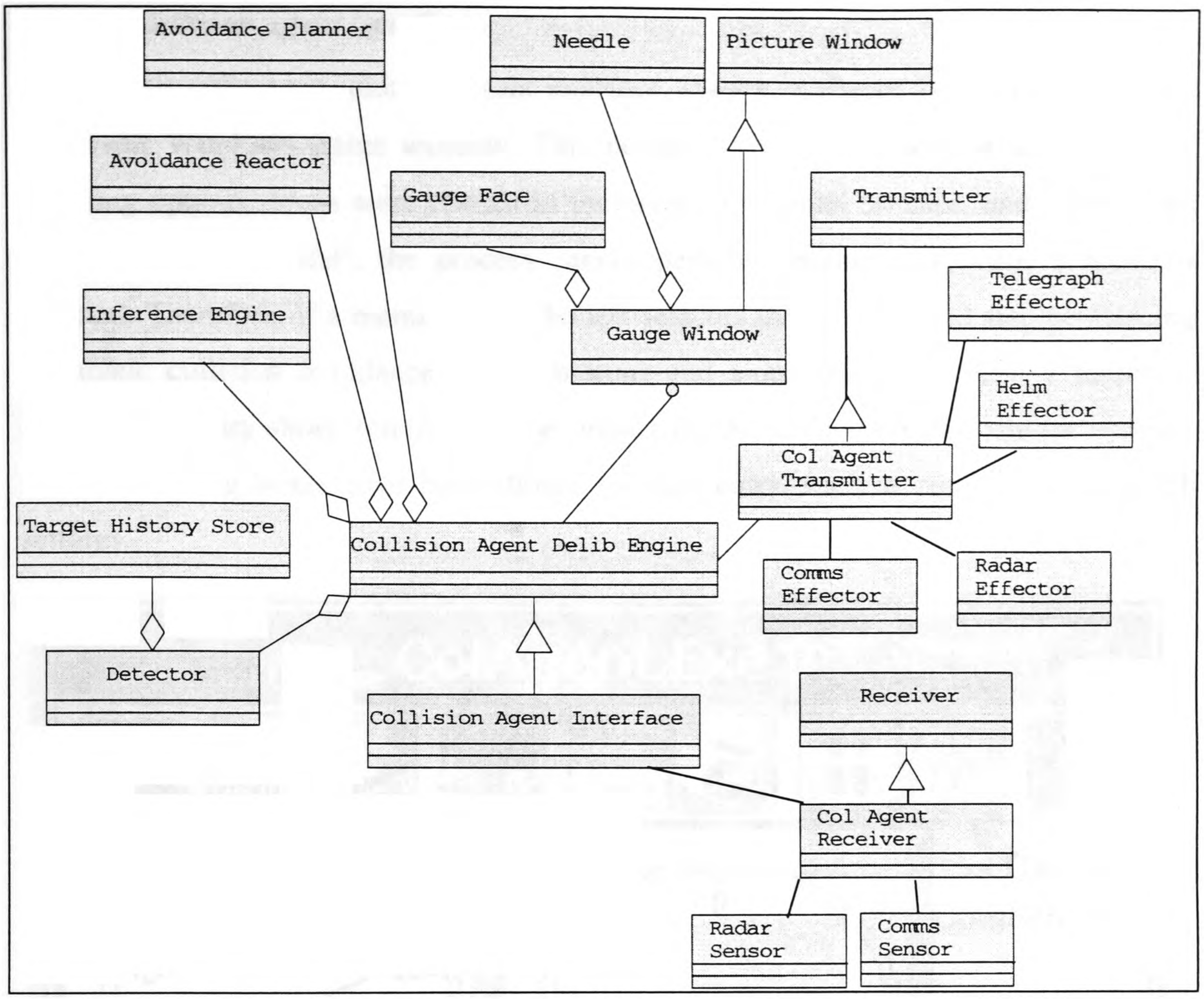


Figure 7-1 The collision avoidance agent architecture

The adoption of this architecture is largely due to the difficulty of planning in a continuously changing environment. Marine collision avoidance, particularly by radar, relies upon accurate historical information about the relative positions of target ships and the course and speed of your own ship. As soon as the navigator makes a manoeuvre the information becomes unreliable; while the own ship course or speed is changing it is not easy to determine whether a target has maintained its own course and speed. Therefore, if the circumstances permit, the navigator may suspend formal planning during a manoeuvre, or when an unforeseen danger presents itself. However, the navigator will continue to monitor the situation ready to react to the changing circumstances. This matches the reasoning of the chosen hybrid architecture.

7.5.1.1 Collision agent interface

This collision avoidance agent interface, shown in Figure 7-2, is provided by a Microsoft Windows frame window. This permits the use of a toolbar and icons for selecting options. Icons were chosen in preference to menus because, under Windows 3.1, menus are 'modal', the process cannot perform another task while a menu is selected. Therefore, if a menu were to be left selected the agent would stop performing automatic collision avoidance. Using buttons and icons the delay when a button is depressed is very short, minimising the impact on the simulation. For similar reasons, modeless dialog boxes have been chosen for data entry, such as course and telegraph settings.

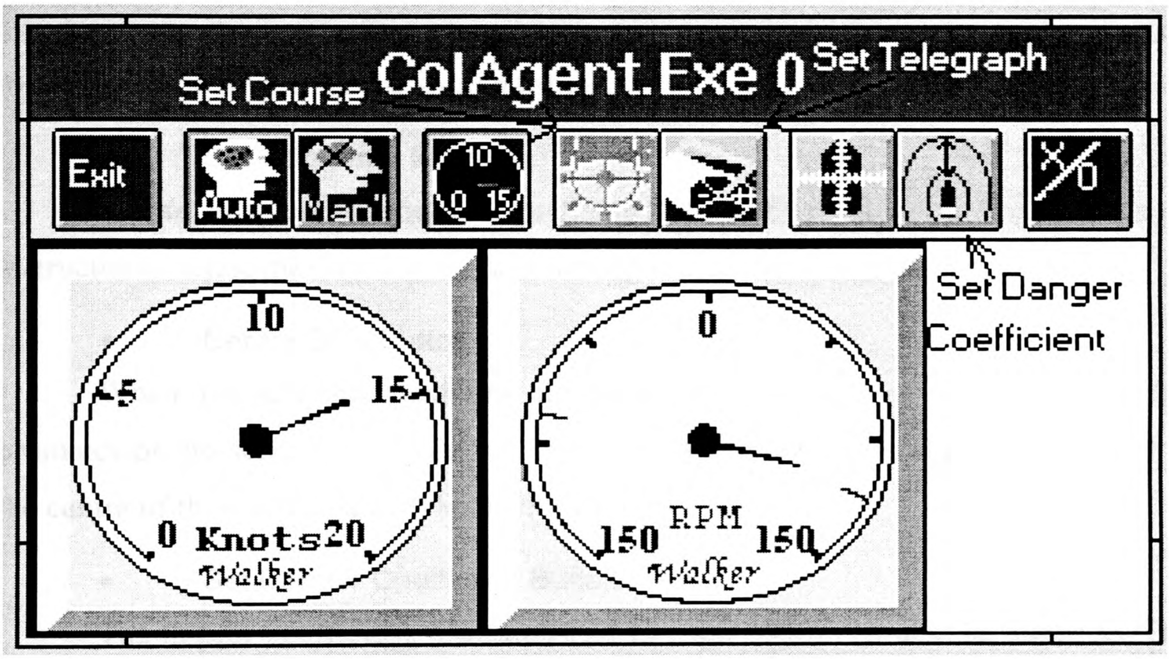


Figure 7-2 The Collision Avoidance Agent Interface

The interface attempts to give the instructor some immediate visual cues about the ship being navigated. This is done by displaying gauges that show the revs and speed; different gauges are displayed for the different ship types.

A very brief description of the buttons provided in the collision avoidance interface, shown in Figure 7-2, is given below.

- Exit Button

The 'Exit' button terminates the visible instance of the Collision Avoidance Agent. As mentioned before, no confirmation warning message is issued.

- Auto/Manual Buttons

Another aspect of the agent interface is the Auto/Manual selection. In auto mode collision avoidance is performed by the agent; the agent monitors approaching traffic and performs collision avoidance manoeuvres as necessary. In manual mode the agent will ignore the approaching ships and follow the desired courses input by the instructor or advised by the track keeping agent.

- Set Course Button

The set course button opens a dialog box for course entry. If both the collision avoidance and the Track-keeping agent are in manual mode this permits the instructor to set a specific course for the ship to follow.

- Set Telegraph Button

The set telegraph button opens a dialog box for telegraph entry. This permits the instructor to adjust the engine telegraph setting from -100 to 100 percent.

- Centre Ship Button

When pressed this positions the instructor station map display at the co-ordinates of the active ship. This has the effect of displaying the ship on the screen at the centre of the visible part of the instructor map.

- Set Danger Coefficient Button

The danger coefficient (Smeaton and Coenen 1990) can also be adjusted. This permits the agent to be given the characteristics of altering course early or late when approaching danger. Typically, a navigator will behave in a fairly consistent manner; either always altering in good time for a hazard or always leaving it late. The navigator of a small manoeuvrable ship will probably alter course later, for a given situation, than one aboard a super tanker. Using this coefficient each agent can be 'tuned' to perform in an appropriate manner.

- Zero Divide Button

This causes an intentional Division by Zero exception. A warning box is displayed and the process terminates. This is used to test the ability of the instructor station to continue running after a fault in another process.

7.5.1.2 Deliberation engine

The deliberation engine is the core of the agent's 'intelligent' processing. At intervals a few seconds apart the agent requests information about the simulation environment. Among other items, the returning information contains the range and bearing of approaching ships. Upon receipt, if enough information is available, the closest point of approach (CPA) and time of closest point of approach (TCPA) are determined. If an approaching ship is too close then the information is sent to the avoidance reactor, normally resulting in immediate action and an alert message to the instructor. If the detected target is at an adequate range the status of the vessel is determined; the type of vessel, whether it is being overtaken., etc. Then the inference engine is called to determine the action to take.

7.5.1.3 Inference engine

The inference engine uses a forward chaining mechanism to parse a set of production rules based upon The International Regulations for Preventing Collisions at Sea (IRPCS 1989).

For example:

```
/ Deal with the situation where the target is nearly directly ahead of own ship
/ and own ship is on the is on the port side of the target
/ therefore passing too close astern
IF TargetAhead AND OwnToPort THEN
        StarboardAlteration
RULEEND
```

The rules are prioritised. As can be seen the result is a general category of action; in this example an alteration of course to starboard. The number of degrees to alter then has to be determined by the planner using mathematical functions or using a 'best guess' from the avoidance reactor.

7.5.1.4 Reactor

The reactor will operate if the approaching ship is: too close and manoeuvring too erratically to permit a mathematical analysis of the situation; too close and a rogue ship that is not obeying the regulations; or too close and too little history exists to determine the course and speed of the approaching ship.

7.5.1.5 Detector

The detector provides the trigonometric functions in order to: determine the CPA and TCPA of a vessel; determine the most dangerous target and set the danger flags used by the inference engine.

7.5.1.6 Planner

The planner takes the past history of each target ship and extrapolates it. The action determined by the inference engine gives a general command, the actual angle of alteration is determined by the planner. This is done for the most dangerous target and then a trial manoeuvre is performed to determine whether the action will result in further danger.

The planner also determines when it is safe to return to course after a manoeuvre. The desired course is used to perform trial manoeuvres upon all the ships that are in the vicinity. If it results in an apparently safe track, for a reasonable time, then a command is sent to the auto pilot to return to the desired course. In manual mode, this desired course is set by the instructor. However, if the track-keeping agent is in auto-mode then it supplies the desired course, updating it to return to the desired track.

7.5.1.7 Transmitter

The transmitter performs the generic DDE client functions; connection, disconnection and sending services. The DDE architecture has been described in chapter 5. This transmitter was briefly described in chapter 6.

7.5.1.8 Collision avoidance agent transmitter

The collision avoidance agent transmitter inherits the basic transmitter and adds functionality to enable messages such as radar information requests and commands to set the ship controls.

7.5.1.9 Receiver

The receiver performs the generic DDE server functions. The DDE architecture has been described in chapter five and this receiver was briefly described in chapter 6.

7.5.1.10 Colagent Receiver

The collision avoidance agent receiver inherits the basic receiver and adds functionality to decode messages such as incoming radar information and own ship information. It also decodes messages about the desired course from the track keeping agent.

7.5.1.11 Needle

The needle is a polygon object. It is actually the same type of object that is used for the ships and the land in the instructor station, as described in chapter 6.

7.5.1.12 Picture window

The picture window is a standardised drawing area. It has been described in chapter 6.

7.5.1.13 Gauge window

The gauge window inherits the picture window and adds a gauge face. The gauge face is a static bitmap resource of the gauge and does not change at run-time. The needle is then superimposed over the face and moved as required. In the original version of MARINES various sizes of gauge were selectable at run time. The object oriented approach made this easy to implement. however, the display became quite cluttered and this was discontinued. Instead, the gauges are now designed to match the ship type.

7.6 Track-keeping agent

The track keeping agent is designed to maintain the desired track of the ship, entered by the instructor. If the ship is moving too far from the desired track then adjustments are then made to return the ship to the track. The agent also calculates the set and drift of any current that the ship is experiencing and applies additional corrections to counteract this. It does this by keeping a history of the track that the ship has recently made good. Additionally, the agent looks ahead to determine if a waypoint is being approached. If this is so, then using its knowledge of the characteristics of the ship it is navigating, it calculates when to commence the turn.

Track-keeping is presently used by companies such as Maritime Dynamics for high speed simulations to test new designs of ports. It has been suggested for integrated ship's bridges (Burns 1995) and forms an integral part of an automatic berthing controller for ships (Kasasbeh 1994). Burns (1992) also suggests a theoretical strategy for providing track keeping and collision avoidance using fuzzy logic.

7.6.1 Track-keeping agent architecture

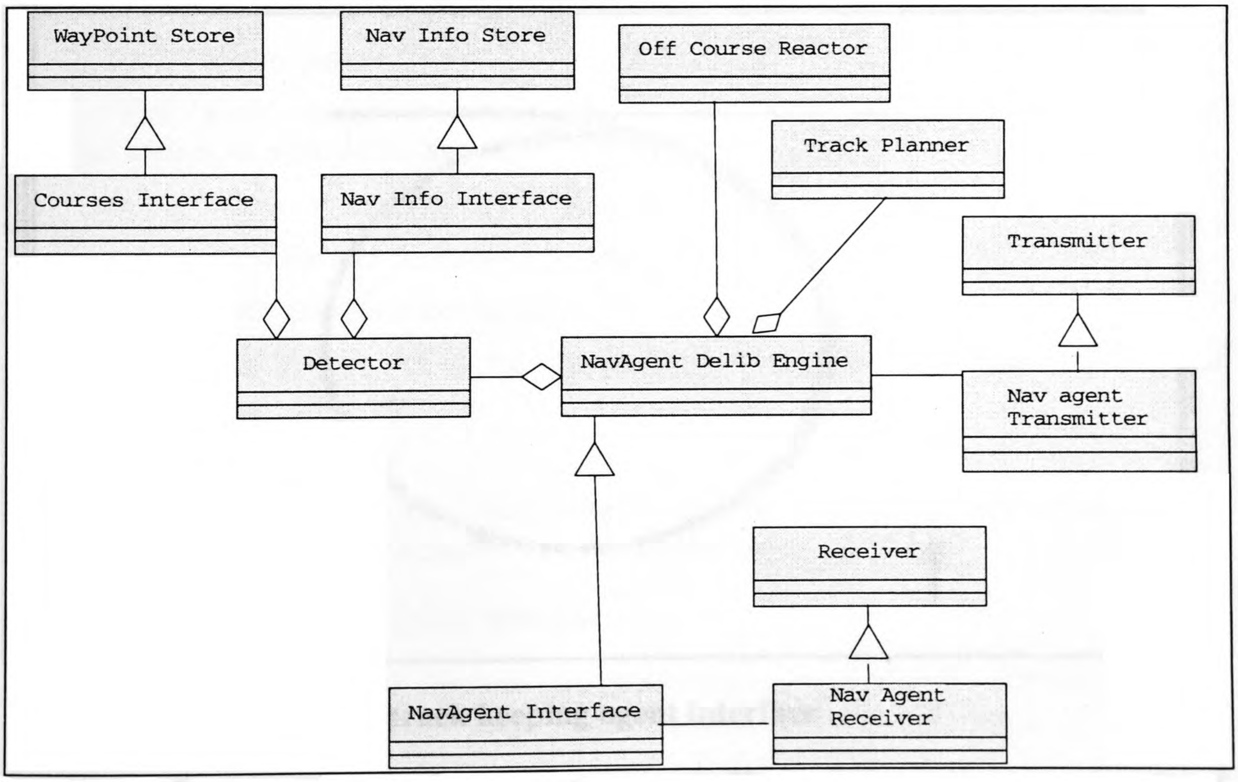


Figure 7-3 The track-keeping agent architecture

The track keeping agent is marginally simpler than the collision avoidance agent. This is because there are no specific rules to apply to maintain the track. Most of the solutions are based upon trigonometric calculations. Once again, there are the reactive and planning components, information stores, and communications systems. However, this time the planner does not include an inference engine.

7.6.1.1 Track Keeping Agent interface

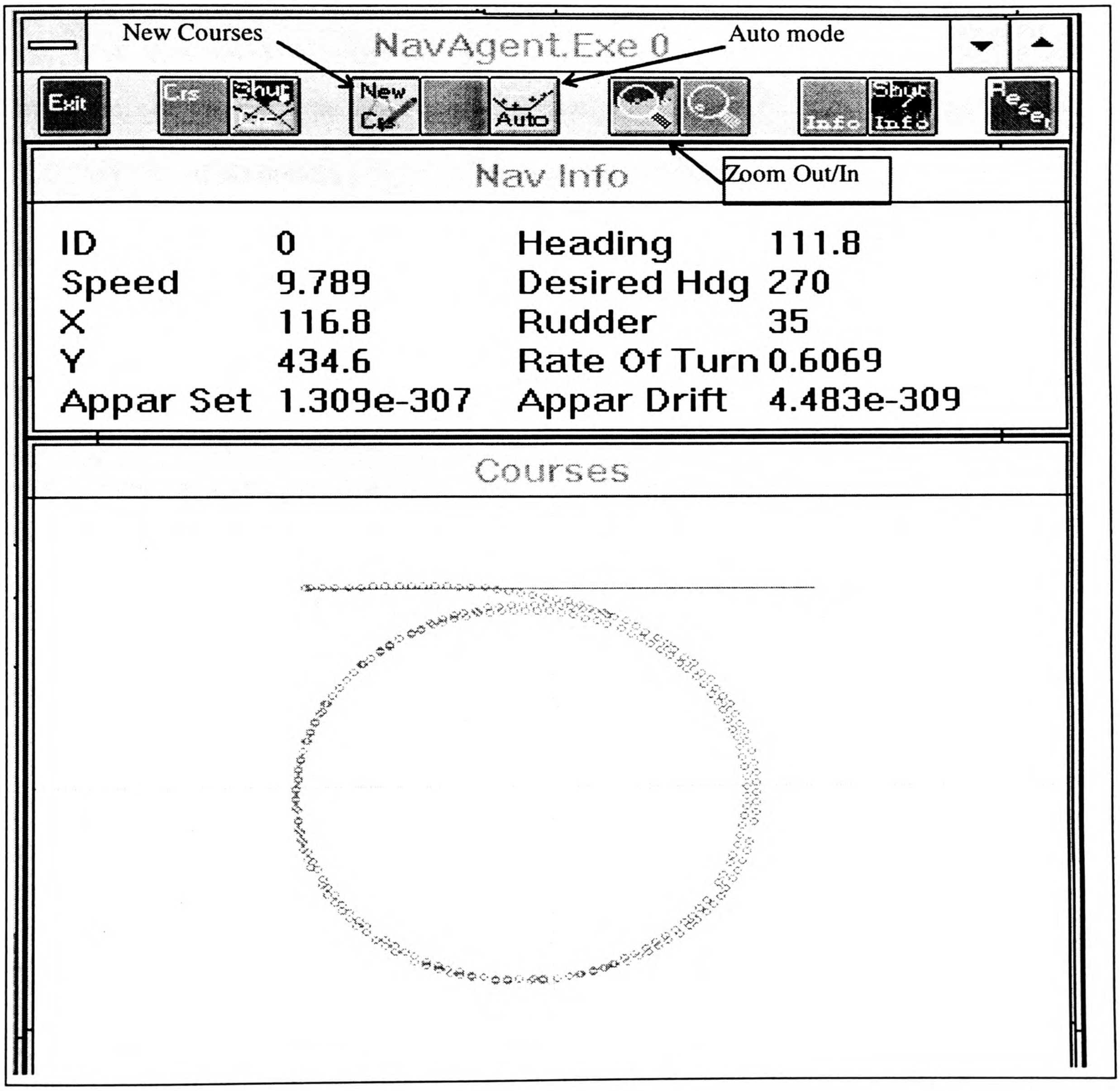


Figure 7-4 The track keeping agent interface

The track keeping agent interface shown in Figure 7-4 provides a crude map of the 'play area' of the simulator in the courses window. Controls are given for allowing the map to be zoomed in and out, although, at present, no controls have been provided for panning around the display. Using this display the instructor can enter courses for the agent to follow. Once the courses have been entered, auto mode can be selected and the agent will try to direct the ship along the desired track. The agent requests new information every few seconds and a small circle is plotted showing the ship position at that time. A short history of the past six positions is stored; and is used to calculate the set and drift of the current.

A very brief description of the buttons provided in the track-keeping agent interface, shown in Figure 7-4, is given below.

- Exit button

The 'Exit' button terminates the visible instance of the Track Keeping Agent. No confirmation warning message is issued.

- Crs and Shut Crs buttons

The 'Crs' button Opens the 'Courses' child window. The 'Shut Crs' button closes the 'Courses' child window.

- New Crs button

The 'New Crs' button permits an additional course to be entered.

- Auto button

The 'Auto' button puts the agent into automatic track keeping mode. New courses cannot be entered in this mode.

- Zoom in and Zoom Out buttons

These increase and decrease the scale of the display in the courses window.

- Info and Shut Info buttons

The 'Info' button Opens the Nav Info child window. The 'Shut Info' button closes the Nav Info child window.

- Reset button

This sends a request to the instructor station to move the ship to the initial position of the first course (Position 1 in the diagram).

7.6.1.2 Courses Interface Child Window

The courses interface permits the instructor to enter new courses for the agent to follow. The Courses window displays the positions of the selected courses.

- New Course Mode

In New Course mode the mouse can be used to drag new courses. Each new course will start from the last point entered. To drag a course the left mouse button is pressed until the desired position is reached. The new point is fixed when the left mouse button is released. If no courses have been previously entered then the first point is positioned where the mouse pointer is when the left button is first pressed.

- Auto Mode

In Auto mode the agent tries to keep the ship on the desired track. The track is shown with parallel lines on either side that represent the boundaries of the track that the ship should remain within. If the agent determines that the ship has reached a boundary then a warning is given to the instructor.

7.6.1.3 Deliberation engine

As in the collision avoidance agent, this forms the core of the agent processing. The deliberation engine sends messages requesting 'own ship' information at regular intervals. The instructor environment returns information about the ship the agent is navigating. This includes the ship's position and its present heading and speed.

7.6.1.4 Detector

The detector provides the solutions to the ship positioning and turning problems. It also provides functions to determine the set and drift encountered.

7.6.1.5 Courses Store

The courses store contains details of each waypoint forming the track which the agent is to try and follow. In v1.07 up to ten waypoints can be entered, giving nine sequential courses to follow.

7.6.1.6 NavInfo Interface

The instructor is able to obtain additional information via the NavInfo Window, which shows: the ID number of the ship that the agent is navigating; the speed in knots that the ship is steaming through the water, the position of the last 'fix'[5]; the actual

[5] A position 'fix' is the place that the ship is calculated to be at the time that the fix is taken. On a real ship the fix will normally be shown in Latitude and Longitude, but, on a simulator they are often held in metres from the exercise origin and this convention is used here. Usually a time is associated with

heading of the ship in degrees; the desired heading set on the auto pilot in degrees; the actual rudder angle in degrees; and the rate of turn in degrees a second. Finally the agent's calculation of the set drift of the current that the ship is encountering is displayed. The set is displayed in degrees and the drift in knots. Note that, in Figure 7-4 no set and drift is present and the agent has calculated a very small value for each.

7.6.1.7 NavInfo Store

The own ship information received from the environment is stored for a few iterations and used by the planner. The information is a superset of the information displayed on the Navinfo Screen, including additional information, such as the height of eye of the observer, the roll, pitch and heave of the ship, the engine revs the time of the message and the simulation rate.

7.6.1.8 Off Course Reactor

In the present version of MARINES the off course reactor is very simple. If the ship deviates further than the allowed distance off track then the instructor is informed.

7.6.1.9 Track Planner

The track planner determines when to make alterations of course and how to apply set and drift to return the ship to the desired track.

7.6.1.10 Nav Agent Transmitter

The Nav agent transmitter inherits the basic transmitter and adds functionality to enable messages such as position information requests and commands to set the ship controls.

7.6.1.11 Nav Agent Receiver

The Nav agent receiver inherits the basic receiver and adds functionality to decode messages such as incoming positional information and own ship information.

---

the fix, however, presently the fixes are taken every few seconds and therefore, in the present track keeping agents no time is displayed.

7.7 Putting MARINES Agents in context with MAS design

As mentioned before, the agents in MARINES have a hybrid architecture; it falls somewhere between the vertical and horizontal layered architectures, although possibly closer to the vertical theory. It can be seen that the main components match those of many other MAS agents: a reactive component; a planning component; information stores; and a control component (Deliberation Engine). However, in MARINES the reactive component is only activated when required, due to an infringement of some domain. This is efficient in keeping processing to a minimum and to some extent mimics the behaviour of a human navigator.

The long term goals that an agent has are pre-set and facts are asserted, such as those describing the relative positions of visible ships. These represent the beliefs of the agents. However, Desires and Intentions are not actively modelled. To some extent the desires and intentions are implicit in the regulations that they observe. For example, a well behaved collision avoidance agent will desire to avoid an approaching ship and intend to follow the collision regulations.

Communications have evolved over several generations of simulations, TileWorld, for example, incorporates a shared data structure containing all the information about the world. A high bandwidth would be needed to implement this as a distributed system incorporating a complex world. NTW (Phillips and Bresina 1991) performs a similar simulation to TileWorld (Pollack and Ringuette 1990) but uses a message passing paradigm; the agents and the world run asynchronously as do the processes in MARINES. In Phoenix (Cohen et al. 1989) and TruckWorld (Hanks et al. 1993) the agents include items such as bulldozers or trucks and a set of primitive operations are provided. However, these simulations update using coarse time steps, this is not suitable for an interactive real-time continuous simulation. The communications between the simulated world and the agents in MARINES differs from most other implementations. The agents in MARINES are based upon human navigators and are provided with an interface that permits interaction similar to that found in the real maritime world. The reason for having agents as navigators rather than as ships is that a navigator will examine the state of the world at intervals a few minutes apart, whereas, the ship position needs to be updated continuously. Therefore, this keeps the interface simple and the communication overhead is relatively low. However, it restricts the use of the MARINES simulator to marine scenarios, unlike Phoenix (Cohen et al 1989) that can simulate a range of problem areas.

On each ship, inter-agent communication occurs directly between the agents, rather than via the simulated world, once again differing from many MAS test bed implementations. This does take a step towards theoretical work on agent frameworks, where researchers have identified communications bottlenecks in architectures organised around a single hub (Steeb et al. 1981).

A pragmatic approach was taken in the design of agents for MARINES and some trade-offs had to be made. For example, an Agent Oriented Programming (AOP) language (Shoham 1993) would have been desirable, but the processing, and implementation, demands for a large number of agents were considerable. In order to keep the communications rapid, a number of fixed messages styles were actually used. However, in line with Shoham's suggestion, primitive operations are supported through these messages. In v1.07 only three primitive operations are used, namely, requesting, informing and ordering. Further primitives such as offering and accepting will be required in the future for more detailed negotiation. Initially, this could improve collaboration between the track keeping and collision avoidance agents. Following this, the concept of traffic management could be introduced, with information from a central traffic controller.

The present design provides agents that are computationally efficient and use a relatively low communication bandwidth. These were fundamental requirements of the MARINES system; it was necessary to run an instructor environment with ten ship models and twenty agents, on a single Pentium processor.

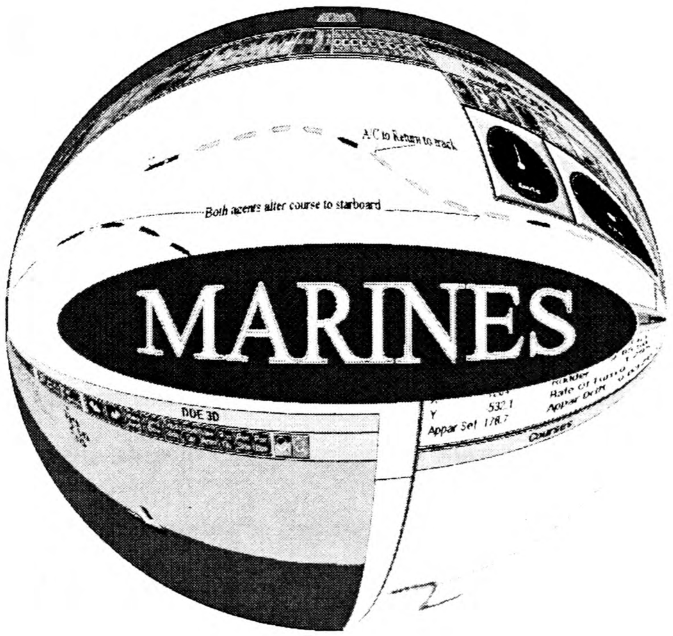
7.8 Summary

This chapter has discussed the construction of the Agents used in MARINES. Some of the components have been derived from classes developed for the MARINES Instructor station. Additional classes have been developed to provide planning, reactive and storage components.

The agent architecture has been put into context with the other researchers' designs. The MARINES agents actually have a hybrid architecture, the components are similar to other architectures, but, unlike other agent architectures, the reactive and planning components are selected by a deliberative control mechanism based upon the level of immediate danger. In some other designs both the planning and reactive components operate in parallel and the results are compared, in others the reactive component is activated first and the calculation is passed up to the planning level if no immediate action is required.

The overall architecture of the MARINES agents provides an efficient implementation that incorporates many of the features that researchers have found valuable.

Chapter Eight



Experiments and Results

8. Experiments and Results

8.1 Overview

A variety of tests have been undertaken to set up the simulator environment and tune the ships. After this, the track-keeping agents have been assessed and then the collision avoidance agents. Various changes were made to the agents to improve their behaviour and then the simulation exercises were performed.

A set of initial experiments were performed in sections 8.2 to 8.4, to ensure that the behaviour of the ship models was consistent with their generic type. Limited tuning of the models was then performed to improve the accuracy of the motion. Following this, the track-keeping of the models was tested to compare the behaviour of the different models when controlled by a navigation agent. Track keeping tests were then performed, in section 8.5 to compare the track keeping when a current was introduced.

In section 8.6 collision avoidance scenarios were set up to determine the collision avoidance agents' capability in single ship collision situations. This was performed with both navigation and collision avoidance agents navigating each target ship, in order to return the ship to course after the manoeuvre.

In section 8.7 an agent using a different rule base was then introduced to show the flexibility of the MAS architecture. This demonstrates the use of specialised agents and also demonstrates that common misconceptions may also be modelled using an agent architecture.

In section 0 scenarios that had initially been performed with no current were re-run with a set and drift from various directions. Therefore, scenarios that were initially, for example, head on, involving risk of collision, became crossing scenarios when the current was introduced, as the ships applied corrections to their courses to maintain the track.

In section 8.9, the ability of the ships to return to track at the correct time was assessed. The return should be soon enough to remain within the boundaries of the desired track and to avoid unnecessary deviations. However, it should not create another close quarters situation.

Although, collision situations involving more than two ships have not been catered for, several simultaneous two ship situations can be dealt with. In section 8.11 exercises containing more ships were considered. Exercises containing first four ships and then ten ships were tried at different simulation rates.

In section 8.12, tests were performed to investigate the MAS architecture. Exercises were performed that introduce and remove agents from the system dynamically at run-time. Further tests were performed to determine what will occur when a software error occurs in an agent at run-time and the degradation of the system when computer resources are unable to meet the processing demands.

The results are assessed further in chapter nine together with the results of the user evaluation.

8.2 Performance in Fast time

The simulator has the ability to run in fast time, V1.07 can operate at up to five times world time on a suitable computer. There are bound to be small discrepancies in the actual model motion due to changes in the integration 'dt' when running at these higher speeds. Some of the tests have been repeated to ensure that the models continue to perform in a similar manner under fast time.

8.3 Ship Models

In order to create consistent models, particularly in fast time or on a slow machine, the implementation has been modified. Delta time (dt) for the ship models has been prevented from exceeding one second (Pourzanjani 1990b); if 'dt' is longer than this then the update is performed in a number of one second steps. This could occur if another process used the processor in Windows 3.1 for too long. It would also occur if the computer was too slow for the simulation rate that was selected. However, in this case the ship motion could gradually fall behind the simulated time. If this occurs then a warning will be issued. Tests are performed in the final section of this chapter to assess the degradation of the system when the computer is unable to support all the processes at an adequate speed.

The ship models are quite rudimentary; they are intended to give approximate turning circles, rates of turn and acceleration for the generic ship type that is being modelled. The models are based upon Newton's second law of motion F = MA, but take no account of the 'added mass' or 'cross coupling' effects between the motions (Pourzanjani 1990a).

Only very limited model 'tuning' has been performed to enable the simulation to run effectively. There are configurable constants such as power and friction which may be adjusted to create more realistic motion. Additionally, it is possible to replace object components, such as the engine, with more realistic models, due to OO design.

The following tests have been devised to obtain a rough guide to the manoeuvring characteristics of the models. These tests give an indication of the maximum speed, the time to accelerate from rest to 50% full speed at full power, the turning circle at full speed and full starboard helm and the distance for a crash stop. No additional tuning has been performed to improve characteristics such as advance[1] and transfer[2] in turn, rates of turn at different helm settings and speeds, speeds at different telegraph settings, etc. This is beyond the present scope of this project. The computer generated target ship models in MARINES are closer to the more complex dynamic models used for the student driven 'own' ship (McCallum 1980), than the simple motion models conventionally used. For example, the target ships in the Mardyn ship handling simulator simply follow the pre-set track at a prescribed rate, without consideration of the environmental effects. The MARINES model creates reasonably realistic ship tracks, however, the subtle nuances of an agent manoeuvring a highly realistic ship model fall into the category of future work.

Figure 8-1a shows the initial test of turning circle for the generic super tanker model, which was performed on a 75Mhz Pentium based machine. The track is the path displayed in the navigation agent courses window during the exercise. It can be seen that the turning circle has too small a steady state radius of turn at approximately 200 metres. Figure 8-1b shows the turning circle of the same ship model with the simulator

[1] Advance is the distance the ship travels forwards when commencing a turn.

[2] Transfer is the distance the ship travels sideways when commencing a turn.

running at five times world time. The scales of the figures are the same and the steady state turning circle is approximately the same. However, there appears to be a small difference in advance and transfer for the model. This may be attributed to at least four factors, the first is the time taken to select alterations by the exercise controller, it being difficult to choose the alteration at precisely the same moment for both tests in the dynamic simulation; this partially accounts for the late turn in Figure 8-1a. The second factor is 'dt' at the different simulator speeds; this may account for the faster initial acceleration causing entry into the turn more rapidly with the longer 'dt' at the faster simulator speed. The third factor, that seems to have the greatest effect, is the precise heading and rate of turn of the ship at the start of the test. Finally, the simulated swell[3] and sea state[4] cause minor changes to the ships heading making the track on each run slightly different.

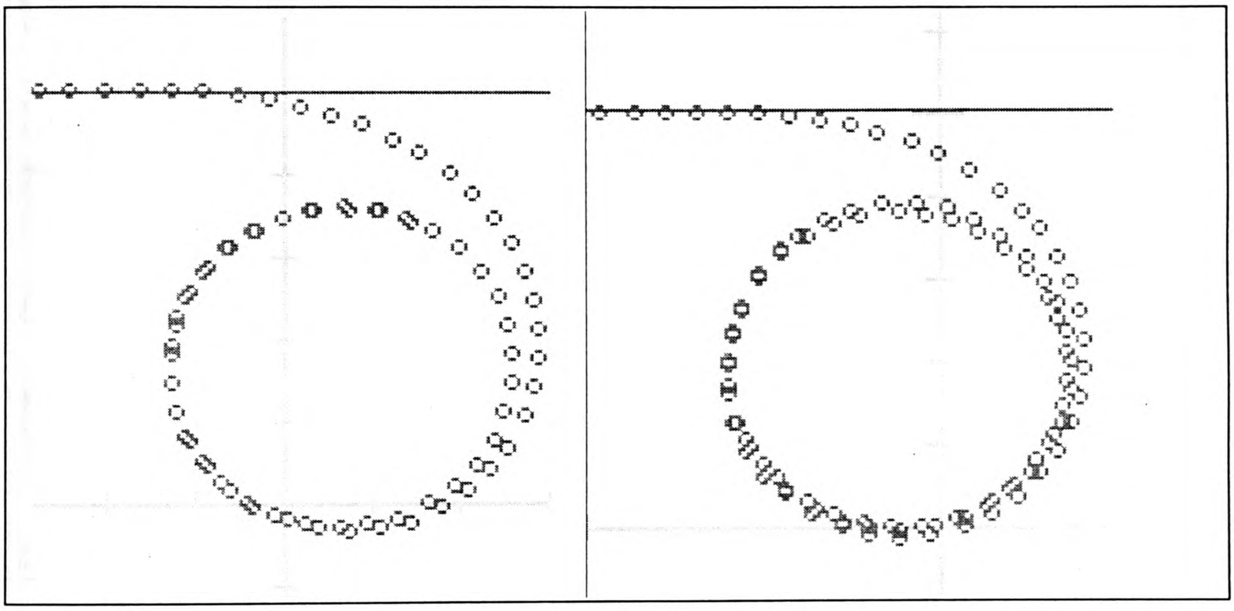


Figure 8-1 a : at world time b: at 5 times world time
Turning circle (Un-tuned Super Tanker model at 10 Knots and 35 degrees stbd helm)

[3] The swell is a measure of the long wave disturbance of the sea. It is usually measured in height from peak to trough and time duration between peaks. The yaw, roll and pitch of a ship are largely due to the swell and the stability of the ship.

[4] The sea state is a measure of the wave height encountered due to wind and other environmental conditions. In open water, the Beaufort scale of wind force is closely connected to the state of the sea.

After this the super tanker model was then tuned to provide a steady state turning circle radius close to the 450 metres expected for a generic super tanker. The test was then repeated three times upon the super tanker turning circle with the simulator running at world time and then repeated three times at five times speed. Once again, the precise heading and rate of turn of the ship at the start of each exercise seemed to affect the transfer, small differences causing up to two or three hundred metres difference. However, the steady state radius of turn was approximately the same in all the tests at around 450 metres. This was considered to be adequate for the altering course requirements for assessing the track keeping and collision avoidance. Figure 8-2a and Figure 8-2b show a repeat of the exercise in Figure 8-1 after this limited tuning.

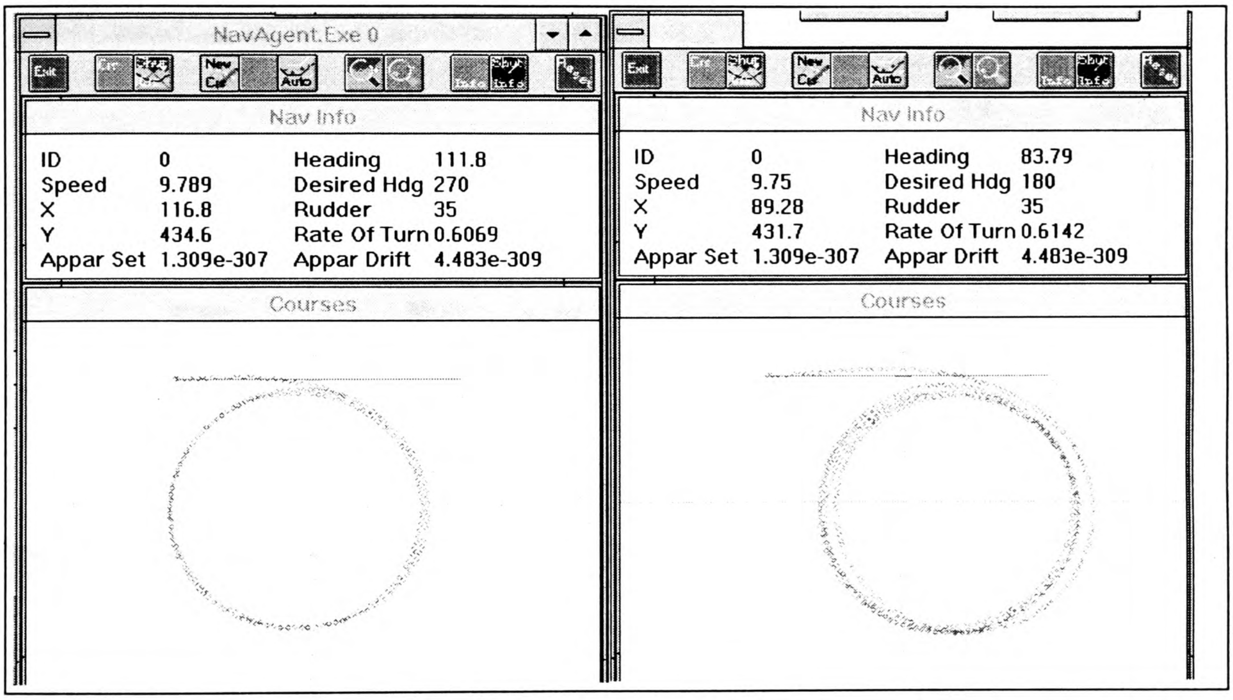


Figure 8-2 a : at world time b: at 5 times world time
Turning circle (Super Tanker at 10 Knots and 35 degrees stbd helm after limited tuning)

In order to obtain more consistent results for the advance and transfer distances it would be necessary to automate the tests and remove the effect of sea state during testing. Running the tests off-line, without the agents or the 3D view could also help to provide this consistency, but, the same results would not be guaranteed as the agents were reintroduced on the simulator. If more accurate models are required in the future then such techniques would have to be employed. However, there would still be a requirement to assess the model in the context of the complete simulation.

The next set of tests determined the stopping distance of the model in a crash stop. The initial stopping distance, before tuning, was considerably too far at 5700 metres. After a limited amount of tuning scenario 2 was re-run three times at world time, each time stopping at around 1710 metres. Similar stopping distances were found when running at five times world time. The turning circle test was then repeated twice to ensure that this remained unchanged. Figure 8-3 shows a stopping distance test for a fishing vessel.

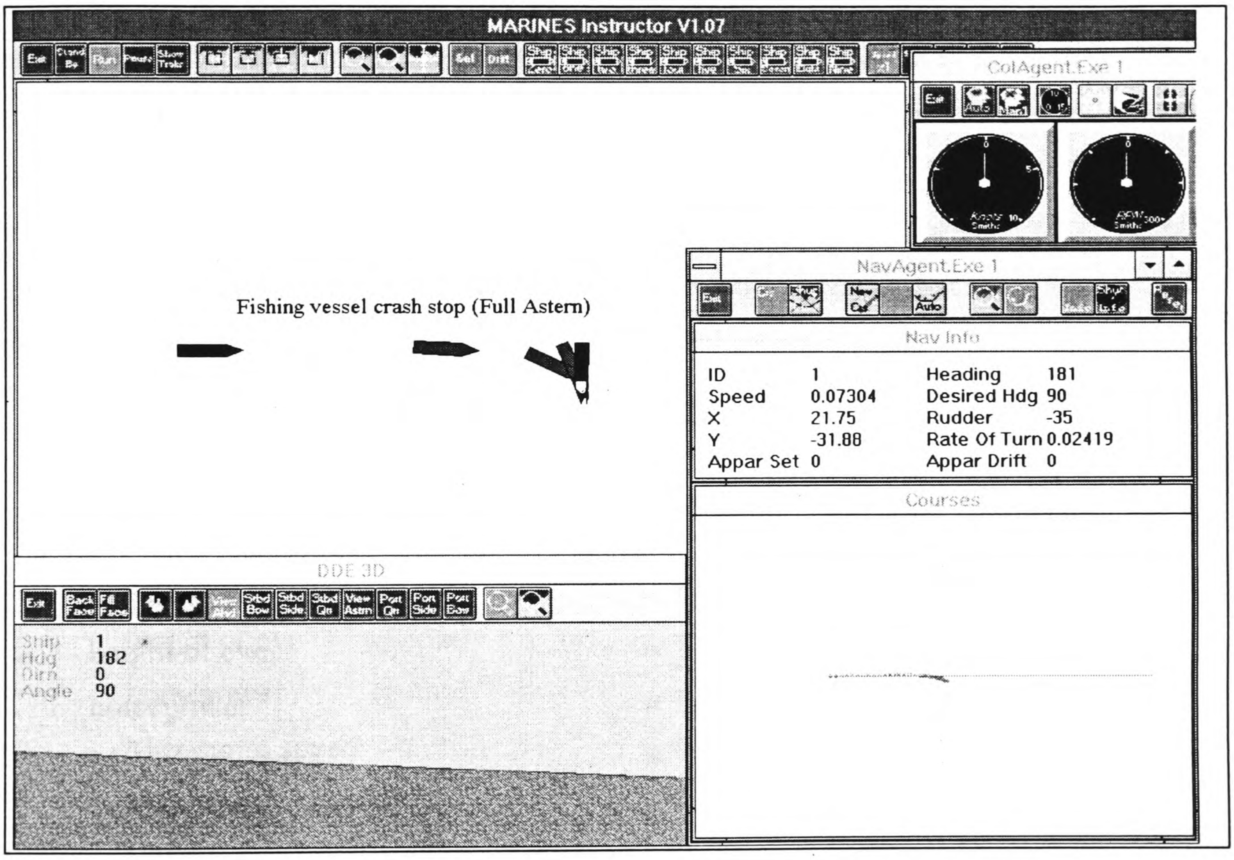


Figure 8-3 Fishing vessel stopping distance

8.3.1 Other Models

Two other models have been produced in addition to the super tanker. The first is intended as a generic products tanker of about 50,000 tonnes, the second model is designed to simulate a trawler, or other small power driven vessel, that is not presently engaged in fishing. The ship is relatively powerful for its size, but has a limited speed due to hull and propeller design, being intended primarily to tow equipment.

Similar experiments have been performed on the tanker and fishing vessel models giving reasonably encouraging results. The average values obtained are tabulated in the next section.

8.4 Results and Model Specifications (After limited model tuning):

NOTE: The radius of turn, advance, transfer and crash stop distances are approximate values averaged from the results obtained from the exercises performed above. The models were tuned until results were reasonably close to the desired values. Desired values for the maximum speed, the crash stop distance and the steady state turning circle are shown in brackets and were obtained as a rough guide from a number of sources, for example McCallum (1980) and Pourzanjani (1990a). As these are generic models it was considered unnecessary to tune these ships to closer values. Information was not discovered for the other values but was considered unnecessary for these unsophisticated models and exercises.

Table 8-1 Super Tanker Characteristics

length	=	300.0	Metres
beam	=	60.0	Metres
displacement	=	300000.0	Tonnes
height of eye	=	35.0	Metres
horsepower	=	30000.0	Shaft HP
Max surge speed	=	7.5 (7.5)	Metres/Second
Max transfer speed	=	2.0	Metres/Second
Max sway rate	=	0.85	Degrees/Second
Maximum shaft revs	=	110	RPM
Minimum shaft revs	=	-85	RPM
Steady state turning radius at 10.0 Knots and 35 degrees starboard rudder	=	470 (450)	Metres
Advance when entering turn at 10.0 Knots and			

35 degrees starboard rudder	=	390 (400)	Metres
Transfer when entering turn at 10.0 Knots and 35 degrees starboard rudder	=	60	Metres
Stopping Distance crash stop from full ahead	=	1700(1750)	Metres

Table 8-2 Tanker Characteristics

length	=	200.0	Metres
beam	=	30.0	Metres
displacement	=	50000.0	Tonnes
height of eye	=	25.0	Metres
horsepower	=	15000.0	Shaft HP
Max surge speed	=	8.0 (8.0)	Metres/Second
Max transfer speed	=	1.5	Metres/Second
Max sway rate	=	1.25	Degrees/Second
Maximum shaft revs	=	110	RPM
Minimum shaft revs	=	-85	RPM
Steady state turning radius at 10.0 Knots and 35 degrees starboard rudder	=	280 (300)	Metres
Advance when entering turn at 10.0 Knots and 35 degrees starboard rudder	=	300 (275)	Metres
Transfer when entering turn at 10.0 Knots and 35 degrees starboard rudder	=	40	Metres
Stopping Distance crash stop from full ahead	=	1400 (1450)	Metres

Table 8-3 Fishing vessel Characteristics

length	=	70.0	Metres
beam	=	14.0	Metres
displacement	=	4000.0	Tonnes
height of eye	=	10.0	Metres

horsepower	=	2500.0	Shaft HP
Max surge speed	=	5.5 (5.5)	Metres/Second
Max transfer speed	=	1.8	Metres/Second
Max sway rate	=	2.25	Degs a second
Maximum revs	=	300	RPM
Minimum revs	=	-300	RPM
Steady state turning radius at 10.0 Knots and 35 degrees starboard rudder	=	150 (150)	Metres
Advance when entering turn at 10.0 Knots and 35 degrees starboard rudder	=	200 (175)	Metres
Transfer when entering turn at 10.0 Knots and 35 degrees starboard rudder	=	20	Metres
Stopping Distance crash stop from full ahead	=	800 (800)	Metres

8.4.1 Assessment of the models

The computer generated target ship models in MARINES provide a model with three degrees of freedom, namely surge, sway and yaw. This is a compromise between the realistic models used for the own ship (McCallum 1980, Pourzanjani 1990a), and the simplistic models of movement conventionally used for target ships in micro simulators. The compromise was made for several reasons: the processing overhead of running ten realistic ship models is high; the lengthy development time for a fully functional model with six degrees of freedom was undesirable; precisely modelling the motion of a specific ship was considered unnecessary for this research; good response to the input devices, such as the mouse and keyboard, was desirable, long periods of intense calculation had to be avoided; shallow water effects[5] and bank effects[6] were unnecessary for this research.

[5] In shallow water ships experience an apparent loss of stability, and manoeuvring capabilities are generally reduced.

[6] Ships experience unusual effects when in close proximity to banks forming the sides of navigable channels. For example, pilots often make use of bank rejection to assist turning in sharp bends in rivers.

The models that were produced appear to perform reasonably consistently and produce acceptable results for target ship manoeuvres. The results show that the desired speeds of the ships and the desired turning circles are achieved. However, under maximum helm the rate of turn is probably too slow and the turn in angle too small. Additionally, the ships do not heel during the turn. The faults are almost unnoticeable on the low resolution visual display in MARINES, especially as most manoeuvres are made at some considerable distance, if the system was used commercially this should be reassessed. The models do offer an improvement over the target ship models that are normally provided on a micro computer based simulator and are considered adequate for the research undertaken.

8.5 Track-Keeping

The track-keeping agent in MARINES calculates alterations of course based upon a limited knowledge of the manoeuvring characteristics for the ship it is controlling. These include the turning circle and approximate values for advance and transfer in turn. The point at which to commence the turn is then calculated using trigonometry; the calculations are based upon the off track distance and the turning circle of the ship. The actual ship manoeuvre is performed by passing a new desired course to the collision avoidance agent and, if the manoeuvre will not produce a close quarters situation, the collision avoidance agent then requests this heading on the auto pilot[7].

The corrections applied to return the ship to the track are based upon the distance off track and the width of the safe water track. The set and drift of the current are then calculated from a history of course steered, log speed and track achieved. The

[7] The auto pilot simulates a Proportional, Integral, Derivative (PID) controller and, as such, tends to apply large helm angles for large alterations of course. This actually helps to maintain the track quite well but would be unsuitable for a large ship performing normal course alterations. Loss of surge speed, shaft vibration, and large angles of heel would, together with their associated results, such as mechanical damage and high level alarms, normally restrict such manoeuvring to emergency situations. Therefore, the agent may, in the future, need the ability to steer the ship manually during large manoeuvres, or a more suitable auto pilot design may be required.

correction of course calculated is then added to the off track corrections to find a desired course to steer.

A number of experiments have been performed to assess the track keeping capabilities of the agents.

Scenarios were devised to check that the agent makes turns in the correct direction in the four mathematical quadrants, when the ship is heading in a variety of directions. In the first exercises the ship had to make a port and a starboard turn of approximately 45 degrees followed by a turn to starboard of approximately 135 degrees. This was followed by a test of a 180 degree turn. The data for this is tabulated in 8.5.3.

8.5.1 Tuning the Auto Pilot

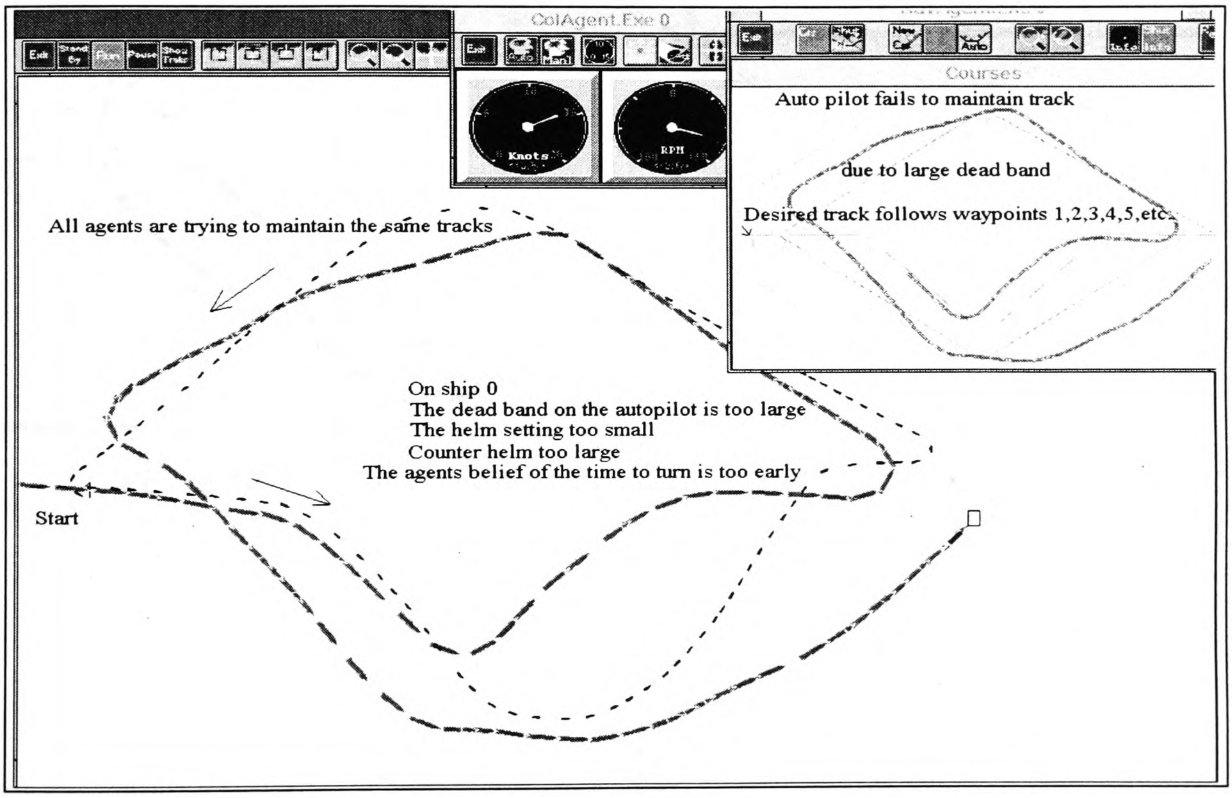


Figure 8-4 Poorly tuned auto pilots (Fishing V/L & Super Tanker)

In a similar way to a real ship the auto pilot has to be tuned to obtain good sea keeping qualities. The settings required for helm and counter helm differ according to the ship model. A number of experiments have been performed on each model while tuning the auto pilot. The settings for helm and counter helm also affect the turning circle when manoeuvring using the automatic pilot, therefore the track keeping agents knowledge about the ships manoeuvring characteristics has to be updated in order to obtain good track keeping qualities. Figure 8-4 displays some of the effects associated with a poorly tuned auto pilot and Figure 8-5 an agent with the wrong manoeuvring information.

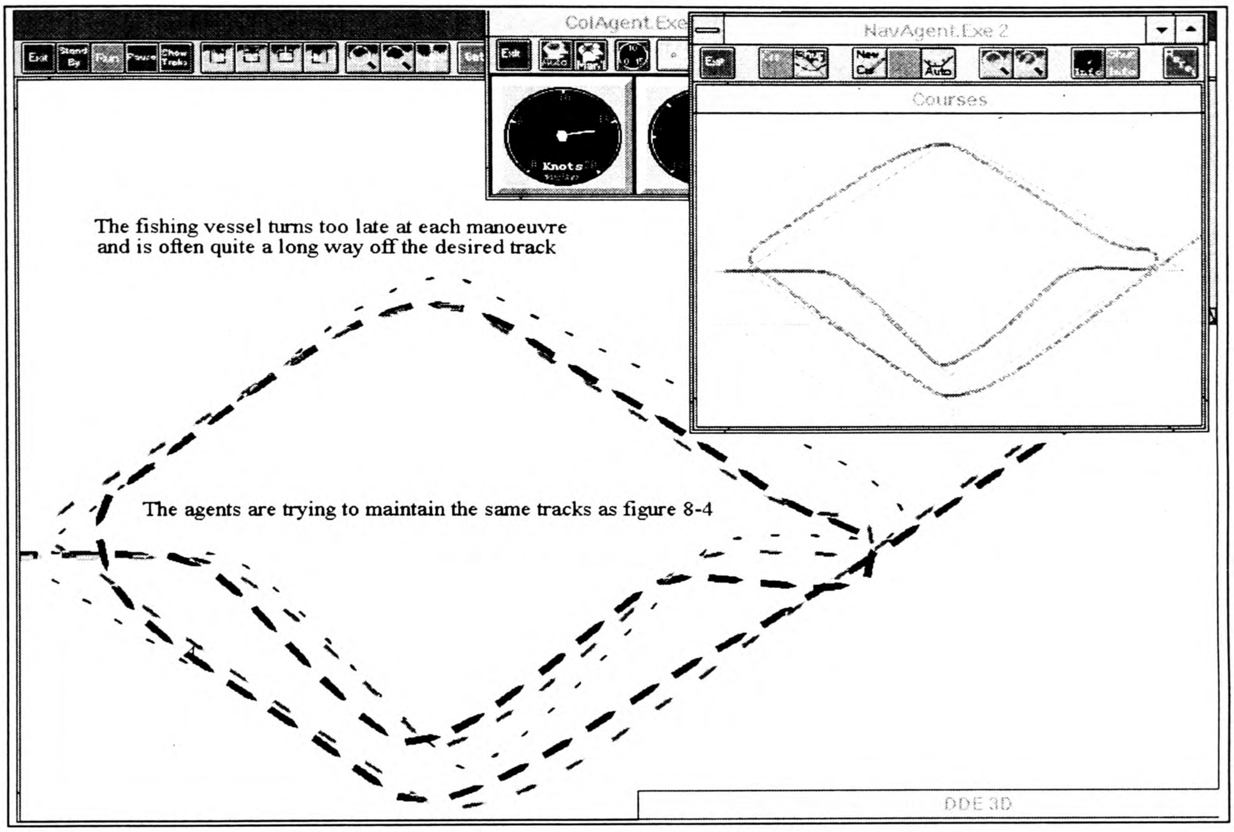


Figure 8-5 Auto pilot tuned, agents with poor manoeuvre beliefs

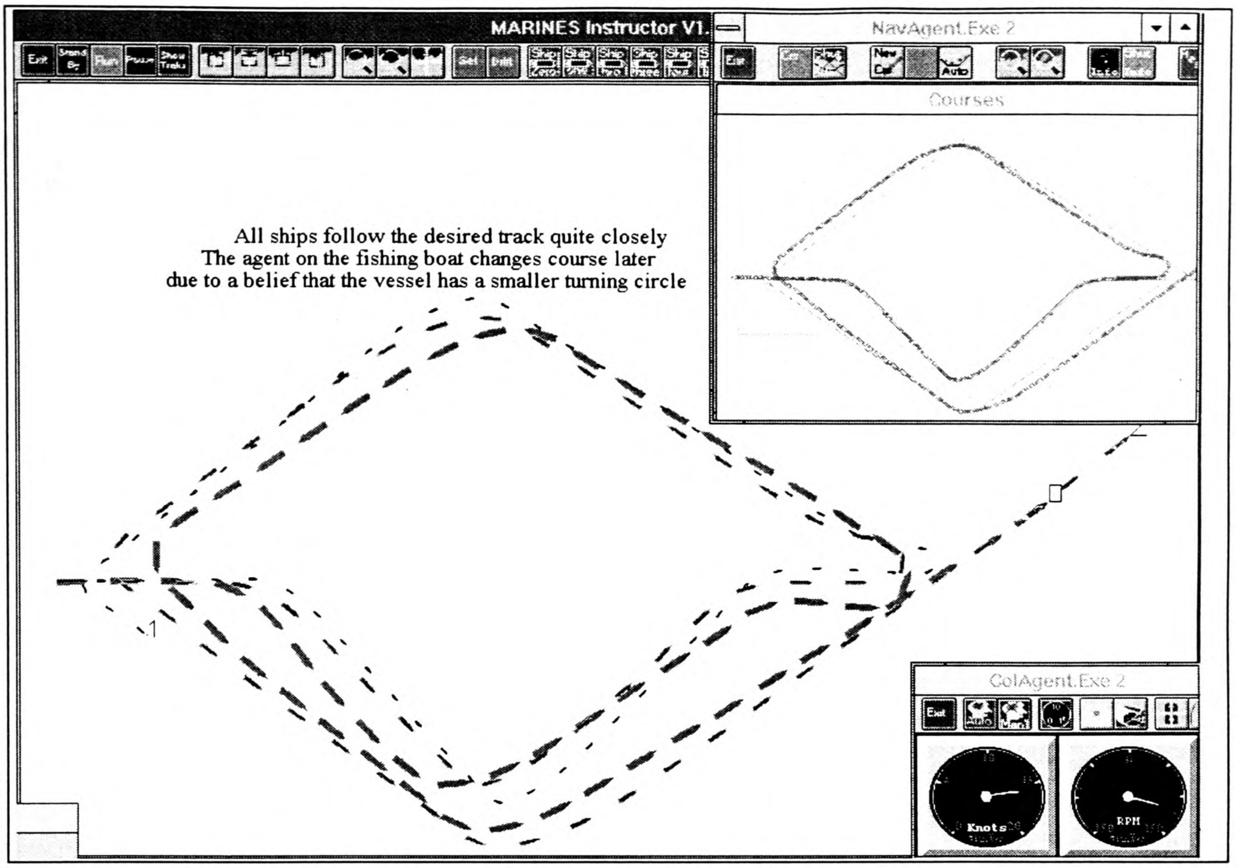


Figure 8-6 Auto pilot tuned, agents with reasonable manoeuvre beliefs

8.5.2 Comparing Ship models

Once the basic track keeping properties had been verified exercises were performed to validate the track keeping characteristics of the different models. Three track keeping agents were each assigned the task of manoeuvring a different ship model over the same desired track. Each ship started the exercise at its maximum steady state speed and the telegraph set to full-ahead.

Exercises were performed to compare the track keeping characteristics of the three ship models. These tests provided comparison of the off track distances and the turning characteristics of the different ship models.

The tracks of the super tanker, tanker and fishing vessel were then compared, it can be seen in Figure 8-7 that the agent on the super tanker commences the turn first and that the larger radius of turn causes the ship to take a route well inside the way point on the tight turns. Additionally, the super tanker takes far longer to settle on to the new track, describing long flowing curves, compared to the relatively tight turns of the fishing vessel and tanker.

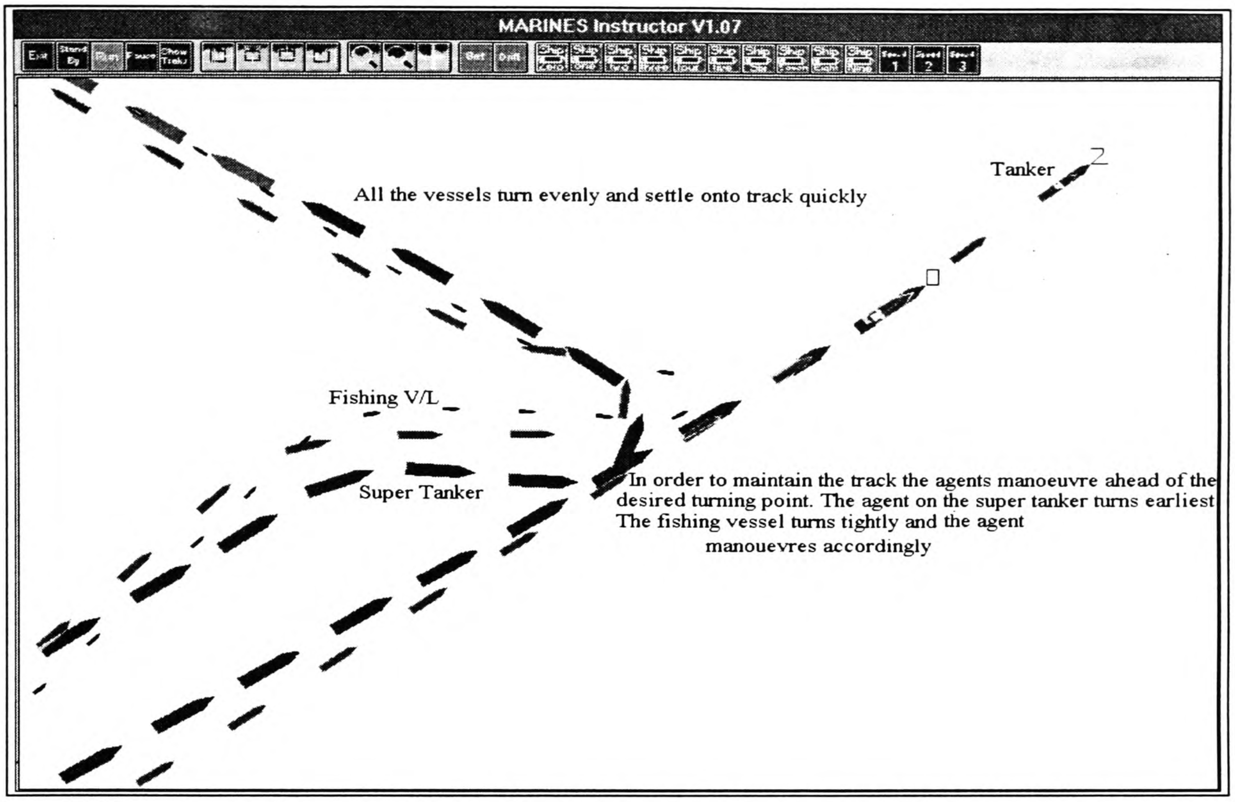


Figure 8-7 Close-up of the track keeping shown in Figure 1-6

8.5.2.1 World Time vs. Fast Time

The exercise was performed to see the similarity to the tracks obtained when the simulation was run at higher rates, and therefore longer ‘dt’. For comparison the same exercise was first run at five times world time and then at world time. For a tanker model, this is shown in Figure 8-8. Note that both tracks are overlaid on the same courses window.

8.5.2.2 Track Keeping Warnings

When the super tanker left the track due to the tightness of the turns a warning was issued. This warning was repeated at three minute intervals. The warning dialog box offers the choice of ignoring the warning or selecting Goto Ship. When the Goto Ship option was chosen the selected ship, in this case ship 0, was centred in the plan view display, the DDE_3D display showed the view from ship 0’s bridge. The agents that controlled ship 0, collision avoidance agent 0 and track keeping agent 0, were also brought to the top. In this way all the available information about the ship is displayed.

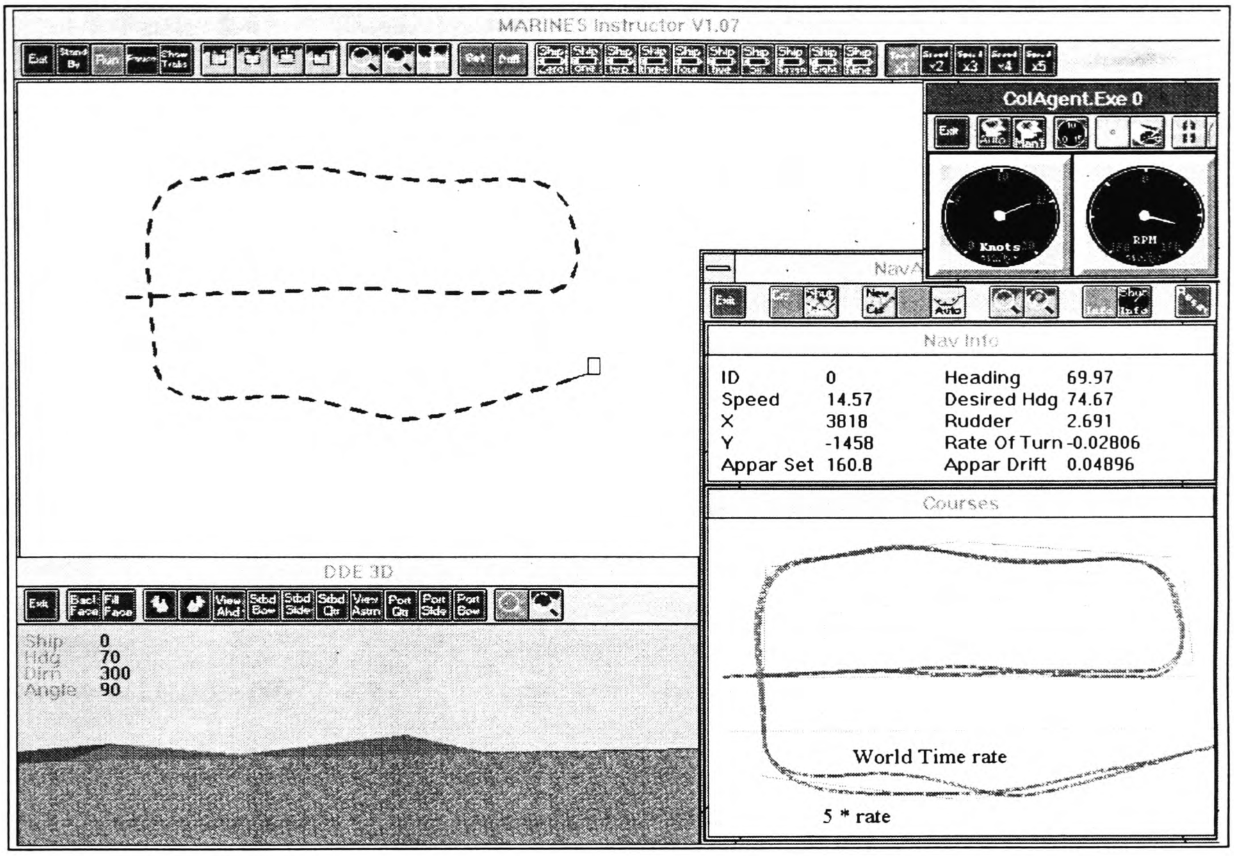


Figure 8-8 A comparison of track keeping at world & 5 * world rate

8.5.3 Track keeping results (after limited auto pilot and model tuning)

NOTE: The results in Table 8-4 to Table 8-6 are obtained under auto pilot steering and do not match the full helm turning manoeuvres of the vessels shown in the ship model section. The Agent Manoeuvre Beliefs are the distances that the agent believes the ship will take to perform a ninety degree turn at full speed, using the auto pilot. The agent uses these values to determine how far before the way point to commence the turn.

The 180 degree turn forces the ship too far off the desired track. This causes a new course to be calculated as soon as the ship approaches the required course. This is similar to the officer of the watch applying corrections before the ship has finally settled on the previous desired course. The agent is free to choose a port or starboard turn for the 180 degree turn. Therefore, the results for the overshoot distance are inappropriate and are marked as not measurable in the following list.

Table 8-4 Super Tanker

	Turn Angle				
	45 degs Stbd	**135 degs Stbd**	**180 degs**	**45 degs Port**	**135 degs Port**
Telegraph Setting (%)	100	100	100	100	100
Ship Speed at start of turn (Knots)	14.57	14.57	14.57	14.57	14.57
Min Speed in turn (Kn)	14.55	14.29	14.17	14.54	14.30
Max Rate of Turn (Degs/Sec)	0.4013S	0.71S	1.074P	0.4527P	1.01P
Max Helm (Degs)	21.08S	35.0S	35.0P	21.33P	35.0P
Max Counter Helm (Degs)	3.29P	3.5P	4.73S	3.25S	3.228S
Max off track Distance (Metres)	250	550	1030	300	450
Overshoot Distance (M)	<20	<20	not measurable	< 60	< 50
Overshoot Heading (Degs)	0.3	1.0	not meas	< 0.7	1.3
Auto-Pilot Helm Setting (0.5..10)	5.0	5.0	5.0	5.0	5.0
Auto-Pilot Counter Helm Setting (0.5..10)	4.0	4.0	4.0	4.0	4.0
Agent Turn Radius Manoeuvre Belief (M)	600	600	600	600	600
Agent Advance Manoeuvre Belief (M)	600	600	600	600	600

Table 8-5 Products Tanker

	Turn Angle				
	45 degs Stbd	**135 degs Stbd**	**180 degs**	**45 degs Port**	**135 degs Port**
Telegraph Setting (%)	100	100	100	100	100
Ship Speed at start of turn (Knots)	15.52	15.54	15.51	15.54	15.54
Min Speed in turn (Kn)	15.49	15.31	15.17	15.51	15.33
Max Rate of Turn (Degs/Sec)	0.4868S	1.598S	1.818S	0.4965P	1.586P
Max Helm (Degs)	20.21S	32.60S	35.0S	18.61P	32.19P
Max Counter Helm (Degs)	4.081P	30.31P	23.41P	3.891S	20.99S

Max off track Distance (Metres)	300	400	628	300	500
Overshoot Distance (Metres)	< 150	< 350	Not Meas	< 150	< 400
Overshoot Heading (Degs)	1.0	15.0	Not Meas	1.5	3.8
Auto-Pilot Helm Setting (0.5..10)	5.0	5.0	5.0	5.0	5.0
Auto-Pilot Counter Helm Setting(0.5..10)	4.0	4.0	4.0	4.0	4.0
Agent Turn Radius Manoeuvre Belief (M)	550	550	550	550	550
Agent Advance Manoeuvre Belief (Metres)	500	500	500	500	500

Table 8-6 Fishing Vessel

	Turn Angle				
	45 degs Stbd	**135 degs Stbd**	**180 degs**	**45 degs Port**	**135 degs Port**
Telegraph Setting (%)	100	100	100	100	100
Ship Speed at start of turn (Knots)	10.69	10.69	10.69	10.69	10.69
Min Speed in turn (Kn)	10.67	10.44	10.35	10.67	10.48
Max Rate of Turn (Degs/Sec)	0.2623S	1.274S	1.732S	0.2681P	1.230P
Max Helm (Degs)	15.88S	25.96S	34.3S	17.21P	25.88P
Max Counter Helm (Degs)	3.004P	35.0P	35.0P	3.169S	39.51S
Max off track Distance (Metres)	180	550	358	220	550
Overshoot Distance (Metres)	180	< 400	Not Meas	< 70	< 450
Overshoot Heading (Degs)	0.8	2.0	Not Meas	2.9	18.0
Auto-Pilot Helm Setting(0.5..10)	6.0	6.0	6.0	6.0	6.0
Auto-Pilot Counter Helm Setting(0.5..10)	5.0	5.0	5.0	5.0	5.0
Agent Turn Radius Manoeuvre Belief (M)	450	450	450	450	450
Agent Advance Manoeuvre Belief (M)	450	450	450	450	450

8.5.4 Assessment of the track keeping

Track keeping been applied to fast time simulation in narrow channels; Maritime Dynamics uses such a system for port design work requiring many trials in differing tidal conditions.

In normal seagoing conditions most ships actually navigate using an auto pilot and it is this type of track keeping that is being simulated in MARINES. As described in 8.5.1 auto pilots often produce quite drastic rudder motion and need to be tuned effectively. Navigators often make a judgement based upon experience, rather than a trigonometric calculation as to when to alter course. Therefore, if anything, the MARINES agents are able to better the track keeping performance of a ship at sea. The agent's performance can of course be degraded by altering the manoeuvre beliefs of the agent.

The results of an agent performing track keeping with an auto pilot are considered adequate for the exercises to be performed. After tuning the models and the agents the ships all followed the desired tracks reasonably closely. Although, at this stage, the effect of collision avoidance upon the track keeping had not been considered.

8.6 Collision Avoidance

In open water and one to one situations the collision regulations (IRPCS 1989) are reasonably clear about the actions and responsibilities to avoid collision[8]. There are a few ambiguities, such as the *"normal practices of good seamanship"*. A well behaved expert system has to resolve the ambiguities without contravening the regulations. Additionally, an expert system for a simulator has to make realistic alterations of course, when compared with the real world. The ships should deviate from their tracks at an appropriate point and return to track without creating further close quarters situations.

In the real world navigators may also make mistakes or suffer from misconceptions about the regulations. Ideally, an expert system created for simulation will be able to emulate common manoeuvring errors upon demand.

---

[8] See appendix A for a description of the collision avoidance problem.

8.6.1 Dangerous target coefficient

A coefficient to determine the most dangerous target (Smeaton and Coenen 1990) was developed at Liverpool John Moores University. This coefficient has been used in MARINES both for selecting the most dangerous target and also for determining when to take appropriate action for that target.

The coefficient relies upon the accurate calculation of CPA and TCPA. However, in a dynamic environment such as that provided in the real world or in MARINES the small changes of direction of the ship's head can make such calculations considerably less than ideal. Therefore, it was necessary to average the results of several exercises. Furthermore, the coefficient is designed to determine which target vessel presents the highest danger at the time of the calculation. Of course, the danger coefficients will change as soon as a manoeuvre is made. Therefore, it is possible that the choice of most dangerous target will change as soon as a manoeuvre is made. Although unlikely, it could be possible that a navigator will alter course for one ship and immediately alter back to the original course to avoid another, and so on. While multiple target avoidance scenarios have not been considered in MARINES, some exercises have been performed to determine which ship will be selected in a number of scenarios. Ideally, trial manoeuvres need to be performed iteratively and new danger coefficients calculated for each trial, this has not been implemented in v1.07.

8.6.2 Overtaking Situations

The overtaking manoeuvre follows Rule 13 (IRPCS 1989). This was performed with a tanker at full speed, of 15.5 knots, overtaking a super tanker in a straight line manoeuvre. The super tanker is manoeuvring at a reduced speed of 10.5 knots, typical of a super tanker running at economic speed. The resulting tracks are shown in Figure 8-9.

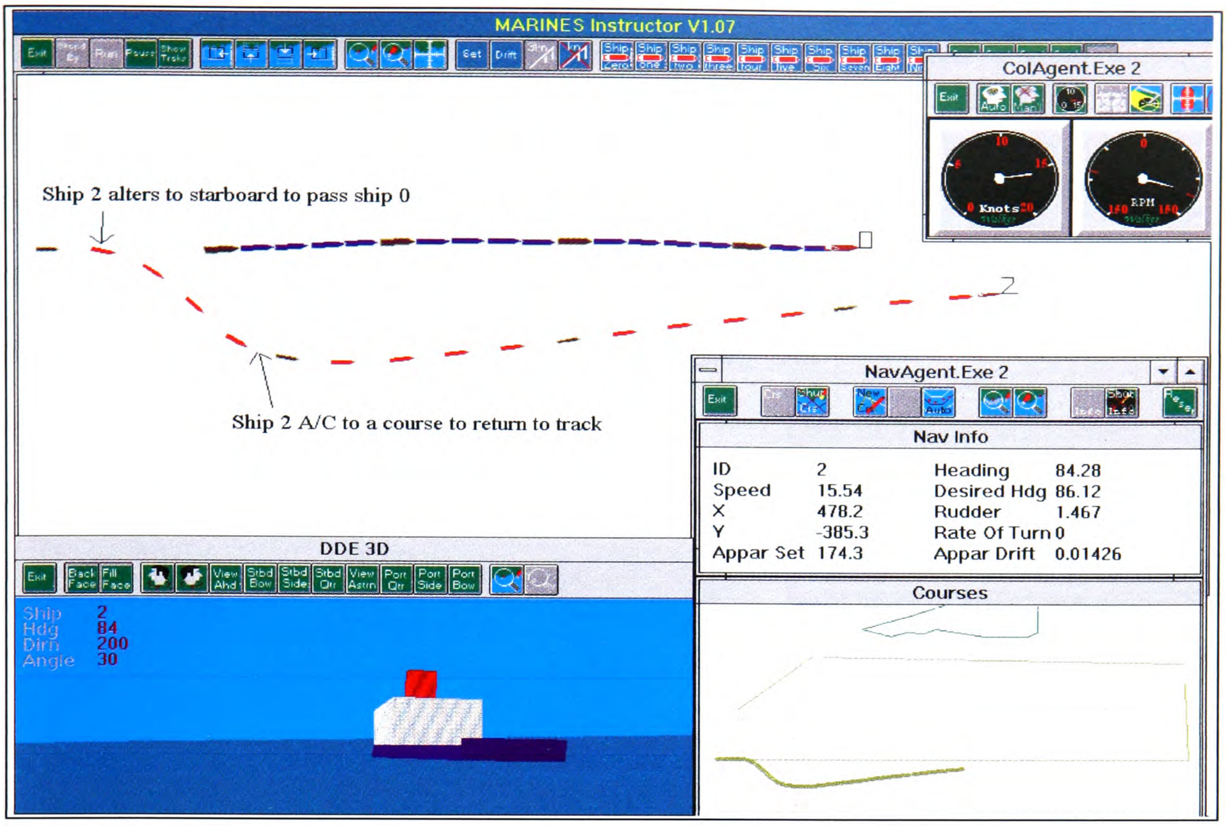


Figure 8-9 A super tanker being overtaken by a tanker

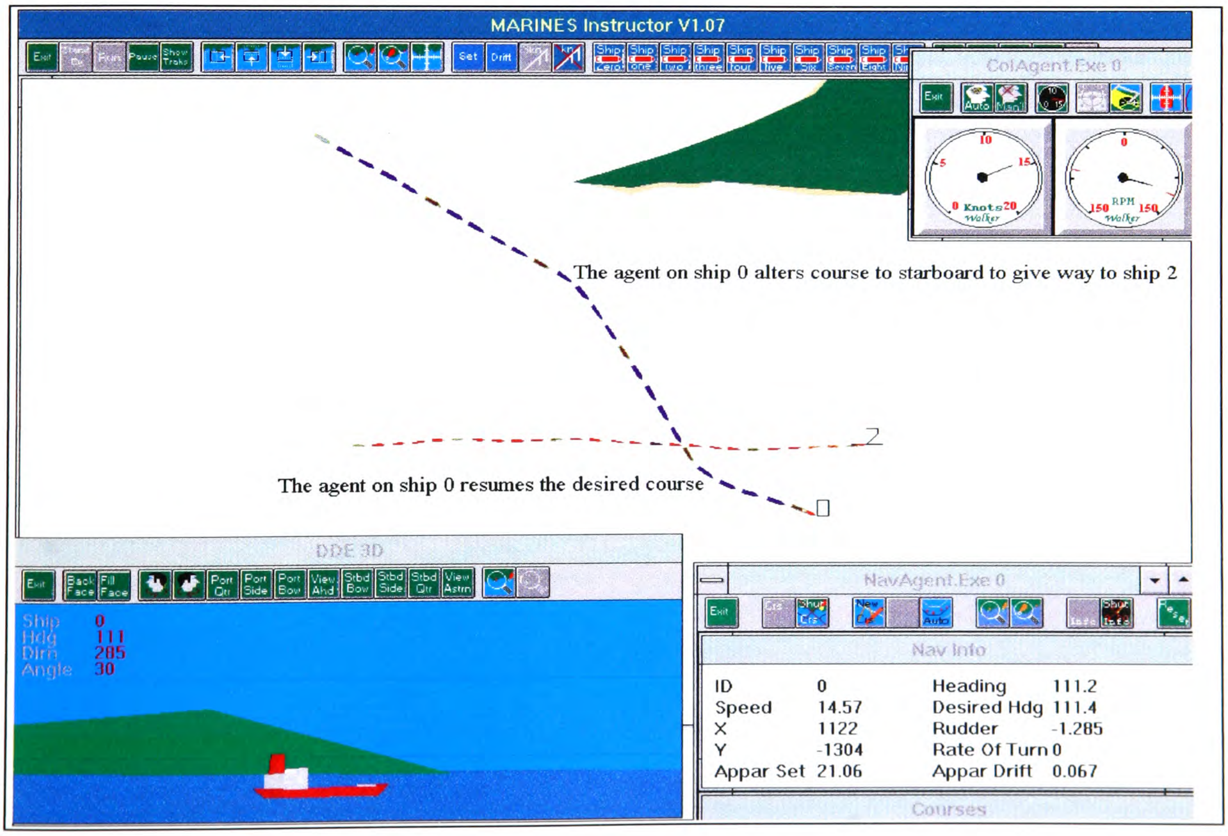


Figure 8-10 Overtaking manoeuvre from abaft the port beam

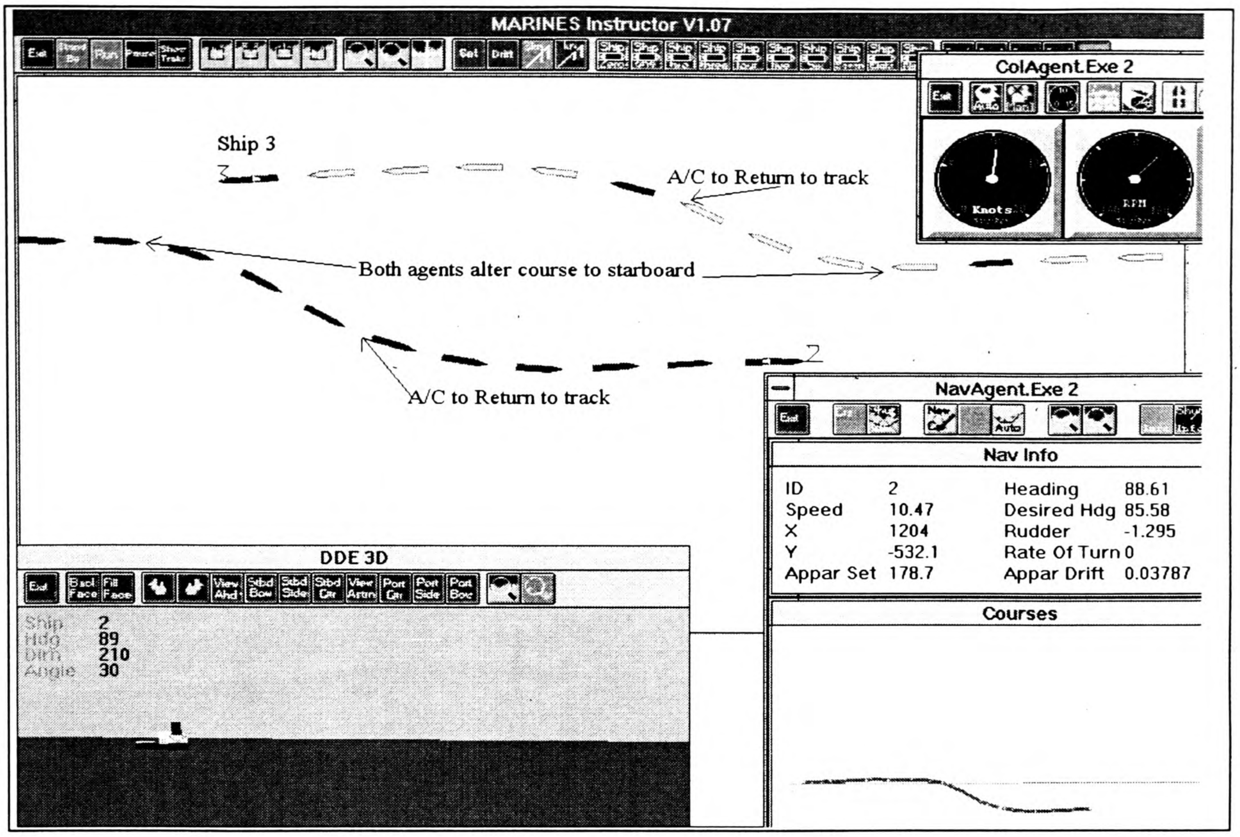


Figure 8-11 Reciprocal tracks without current

Further overtaking exercises were performed with other vessels and speeds. The manoeuvres were successful, however, it would be better if the collision avoidance agent changed course to overtake a little sooner and made a less severe manoeuvre. The return to track was reasonably smooth and effective.

8.6.3 Head-On Situations

Action in a head on situation between two power driven vessels is described in Rule 14 (IRPCS 1989). The exercises in this section assess the point at which a ship will begin a manoeuvre in such a head on situation at different speeds of approach and different threshold values for the coefficient.

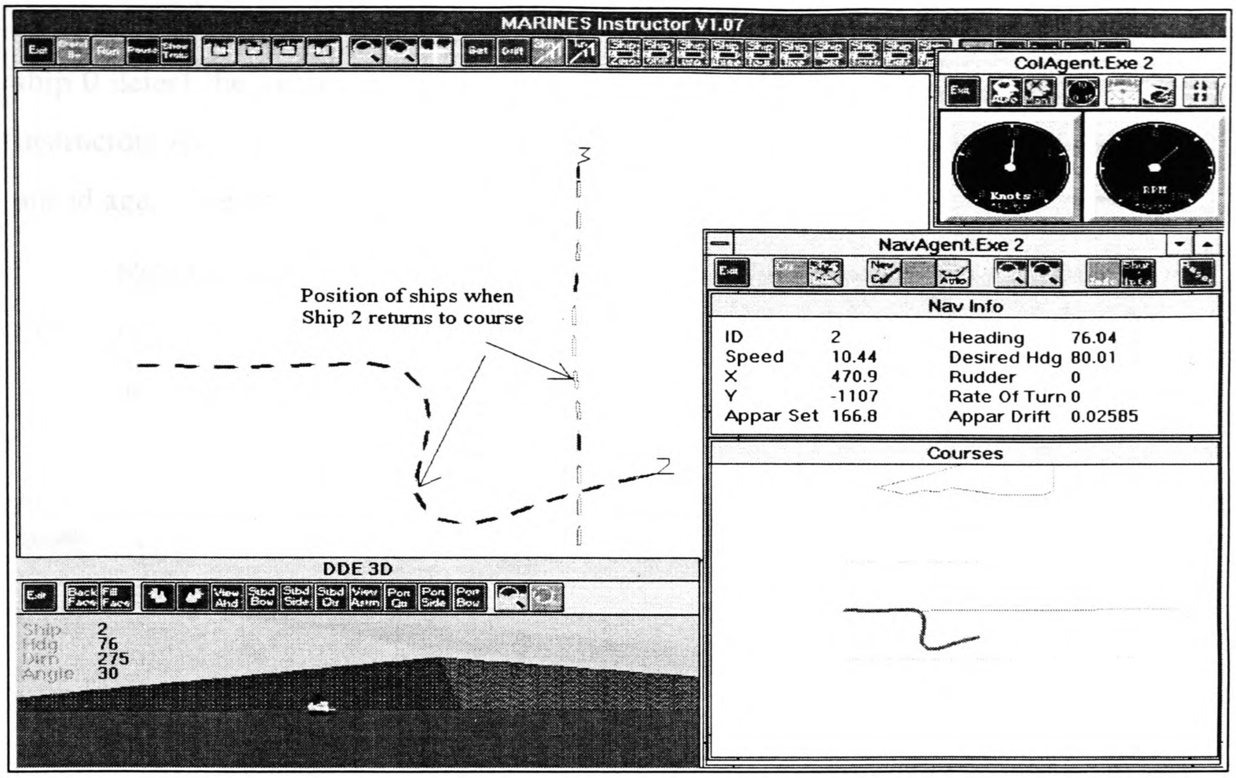


Figure 8-12 Right angle crossing situation with no current

8.6.4 Crossing Situations

In Figure 8-12 the agent on ship 2 desires to make good a course of 090(T) and the agent on ship one a course of 000(T). Both vessels are navigating at approximately 10.5 knots. Once again, two target ships are considered and an identical ship model is used for both ship 2 and ship 3. Under rule 15 (IRPCS 1989) the agent on ship 2 alters course to starboard passing around the stern of ship 3.

8.6.5 Unexpected situations

In Figure 8-13 the track keeping agent attached to ship 0 alters course to port to follow the desired track at the end of the headland. The collision avoidance agent attached to ship 1 has been monitoring ship 0 and detects danger of a collision, this agent alters the course of ship 1 to starboard at the same time as ship 0 alters to port. Shortly after this both agents detect the new danger of collision. Due to the close quarters situation that is developing both agents make reactive manoeuvres. The collision agent attached to ship 1 puts the telegraph full astern to stop ship 1. The collision agent attached to ship 0 alters course to starboard. Action by agents on both ships has avoided the collision, however the track keeping agents attached to ship 1 and ship 0 detect the vessels leaving the desired tracks and send advisory messages to the instructor. At this time the instructor intervenes and puts the telegraph of ship 1 full ahead again, the ships then gradually return to their desired tracks.

NOTE: This exercise was performed before the addition of the buttons for fast time running, or full model tuning. Since then it has not been possible to precisely re-create the effect.

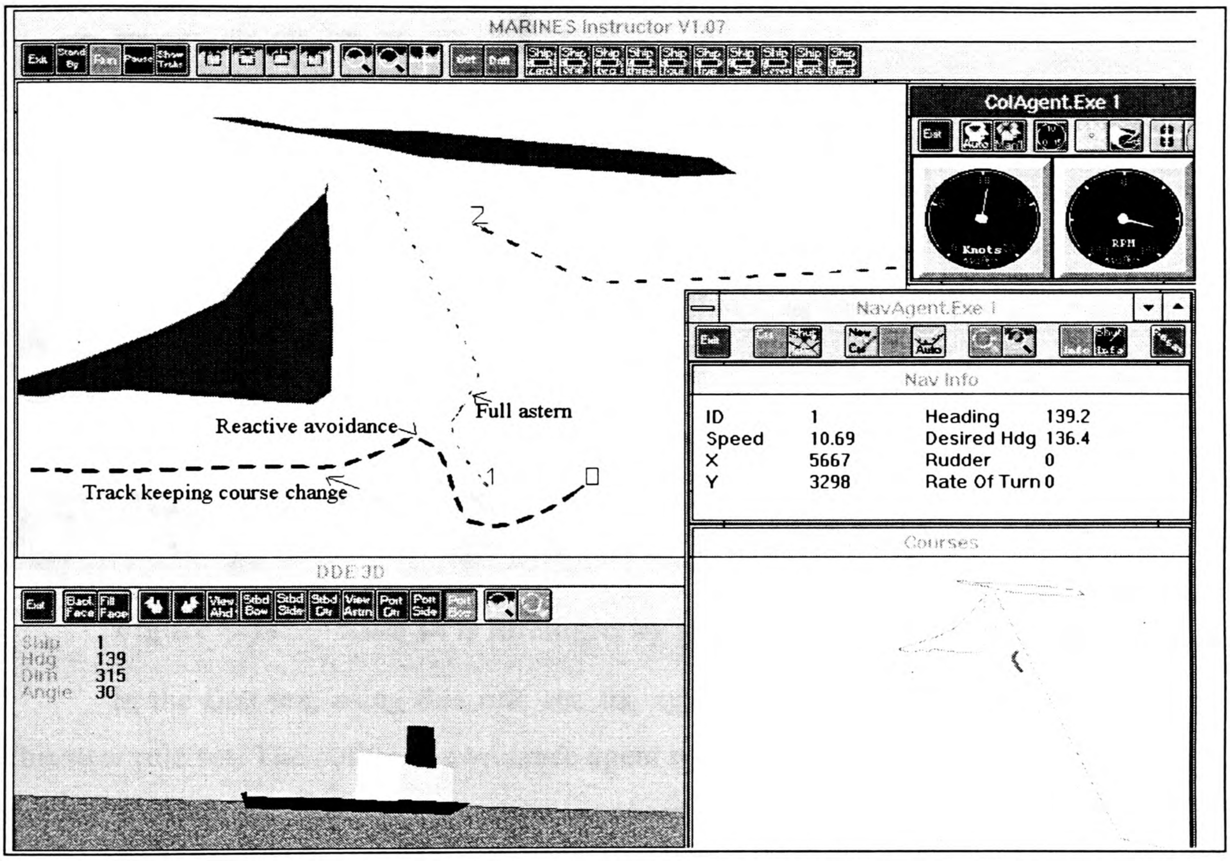


Figure 8-13 Unexpected Situations

8.7 Testing a different rule set

A second rule set was introduced to cause the agent to alter course to port when a head on situation occurs where the ship is passing close down the starboard side. In practice, altering course to port in this situation is a common infringement of the collision regulations (IRPCS 1989). A navigator sees another ship on a reciprocal course that is already passing down the starboard side, but will pass a little close. The navigator chooses a small alteration to port, rather than a bold alteration to starboard to obtain the necessary safe domain. However, should the navigator aboard the other vessel follow the rules and alter course to starboard then a close quarters situation may result.

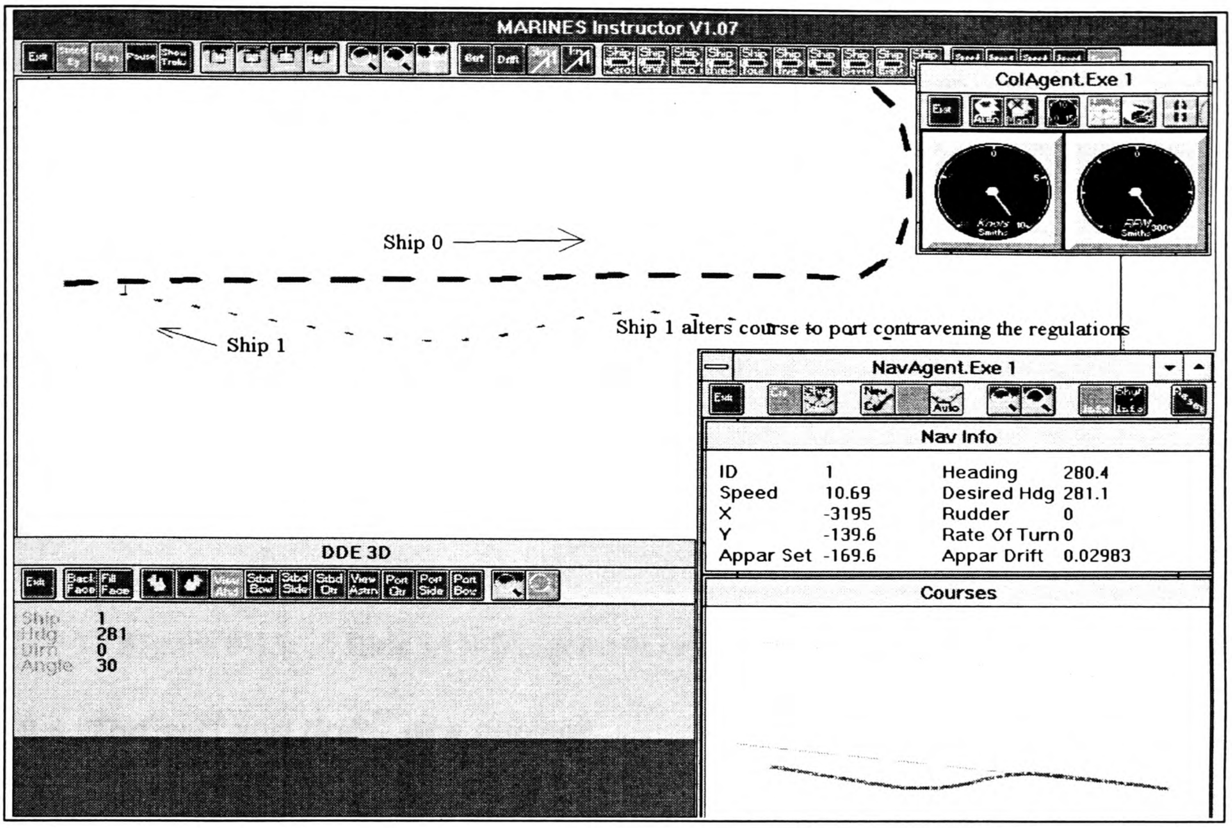


Figure 8-14 Rule 14 is infringed by an agent on the fishing V/L

In the first test using this rule set, the agent controlling the fishing vessel used this new rule set. The collision avoidance agent navigating the super tanker (ship 0) was not set to automatic control. The results are shown in Figure 8-14.

Three further tests were performed using the same head on scenario. However, in these exercises, the super tanker was controlled by an agent with the original rule set. The setting for the dangerous target coefficient of the agent attached to the fishing vessel was altered in each to obtain different interactions, An example is shown in Figure 8-15; the coefficient on both vessels is set to 7.5 and the fishing vessel makes a small alteration to port at the same time as the super tanker alters course to starboard resulting in a dangerous close quarters situation.

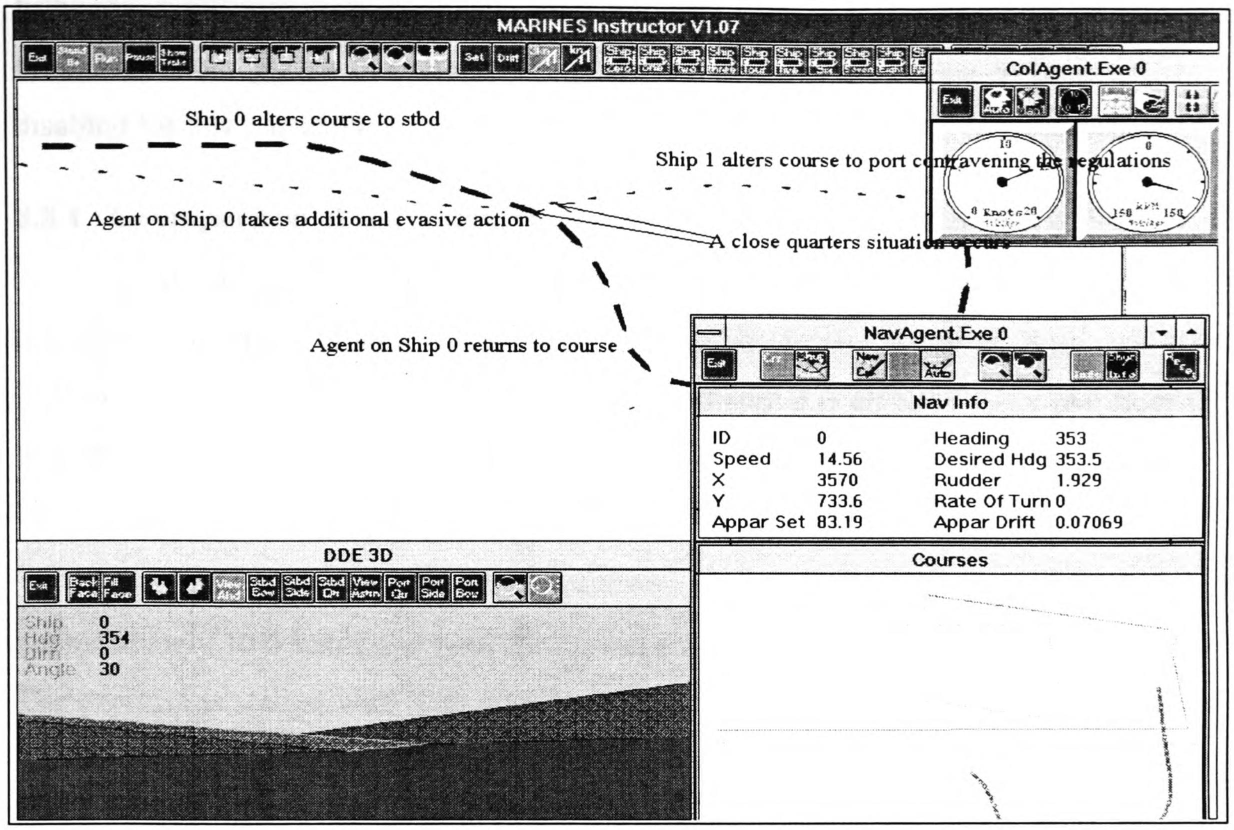


Figure 8-15 Rule 14 infringement causes a close quarters situation

8.8 The set[9] and drift[10] of a current

The effects of land stabilised[11] and sea stabilised[12] target ship motion are now considered. The following exercises have been devised to highlight the differences when a current is introduced and show how the MARINES simulation performs with

[9] The 'set' of a current is the direction that a ship is moved by a current.

[10] The 'drift' of a current is the distance that a ship is moved by a current. The actual speed at which the current flows is termed the 'rate' of a current. The interface button in MARINES is termed 'Drift' to avoid confusion with the simulation rate; it is actually accepting the drift per hour or rate of the current.

[11] If an object in the sea is unaffected by environmental conditions such as the set and drift of the current then the course and speed made good over the land is the same as the speed and course steered. The motion is relative to the land, i.e. land stabilised motion. Note that, in the real world, an object would need a physical connection to the land in order to behave in this way. The land stabilised track of a ship shows the true historical positions of the ship.

[12] A ship manoeuvring at sea is usually affected by the prevailing current. Therefore, ignoring the effects of propeller slip, etc., the motion is relative to the body of sea water (sea stabilised) not the surrounding land. A vector for the set and drift of the current has to be applied to the speed through the water to obtain the speed made good. On a simulator the own ship is normally affected by the current, while the computer generated ships seldom are. In the real world collision avoidance between ships that are under way is sea stabilised.

fully sea stabilised target ships. For demonstration purposes, two further buttons have been added that permit the effects of the set and drift of the current to be enabled or disabled for any particular target ship.

8.8.1 A comparison of sea stabilised and land stabilised motion in MARINES

In the exercises shown in Figure 8-16 to Figure 8-21 two target ships are considered, an identical ship model, that of the tanker, is used for both ship 2 and ship 3. However ship 2 is unaffected by the current and ship 3 is affected. Note that in order to permit the close quarters situations to be demonstrated the collision avoidance agents have not been enabled for these exercises.

In Figure 8-16 the telegraph on each ship is set to 40% power, to give a speed of approximately 10.5 knots and both auto pilots are set with a desired heading of 090 (T). The exercise is started with both vessels in the same position and they are permitted to run freely without any track keeping corrections. Ship 3 is pushed northwards at three knots, while ship 2 makes good the track it steers.

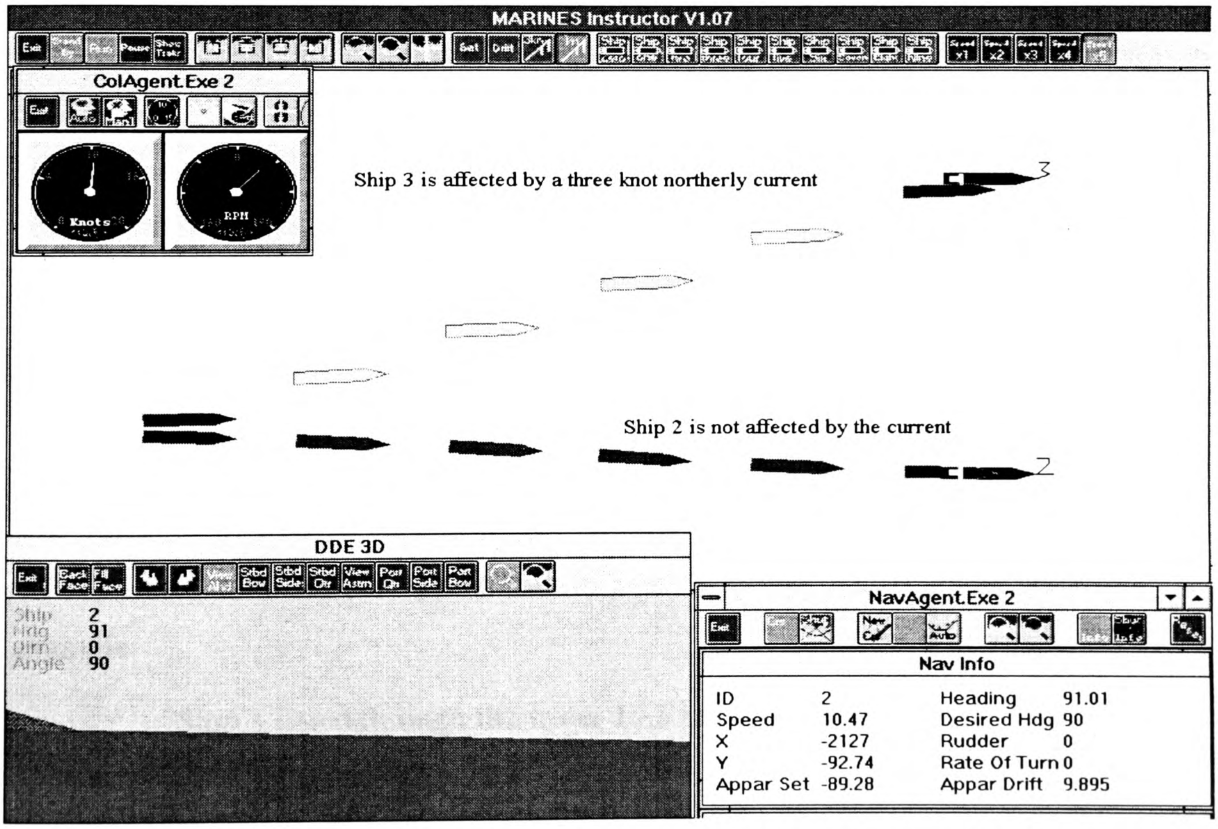


Figure 8-16 Ship 2 is land stabilised and Ship 3 is Sea stabilised.

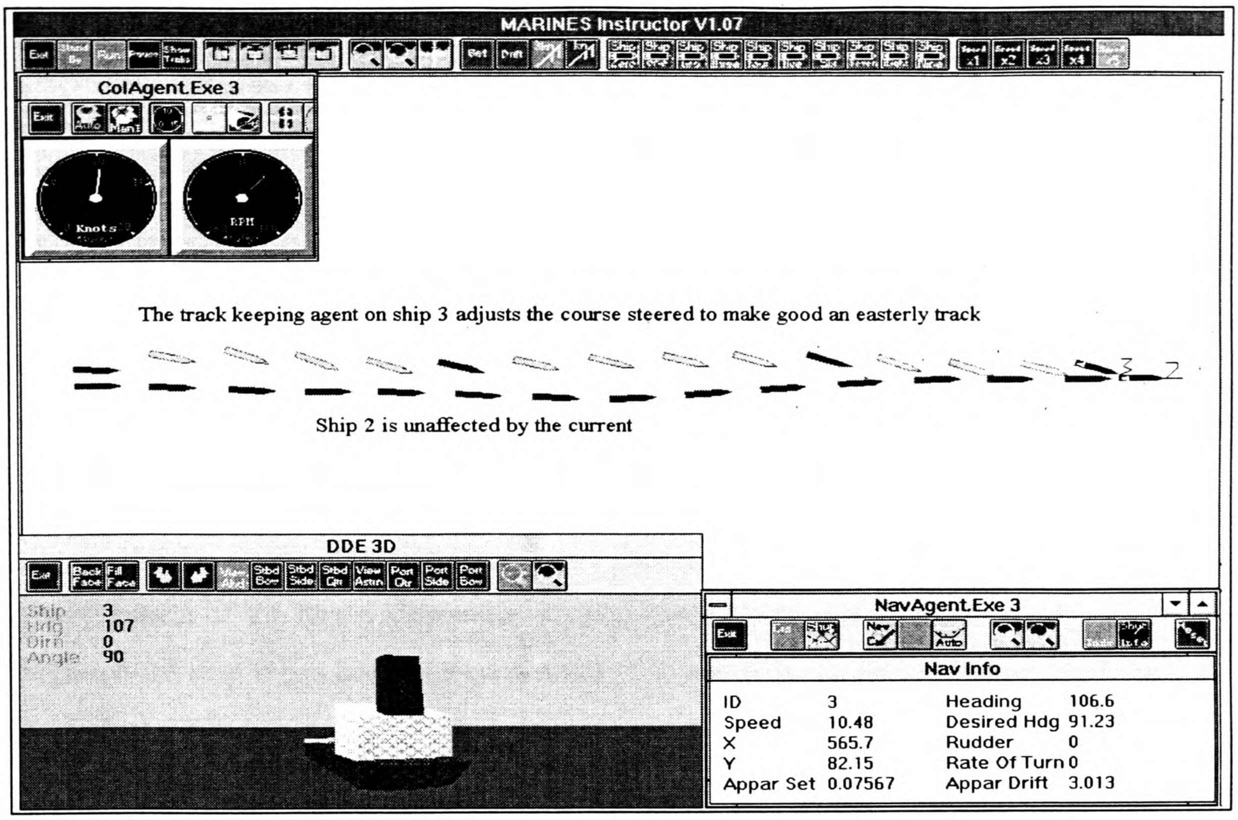


Figure 8-17 Both ships are trying to make good a desired track of 090.

In Figure 8-17 the previous exercise is repeated, in this case the agent navigating on ship 3 is in automatic track keeping mode. The agent tries to overcome the current by applying corrections to the desired heading which is set on the auto pilot. From the start position on the left of Figure 8-17 the ship is initially pushed northwards by the current. After the agent has had time to detect the set and drift, a new heading is applied and the ship begins to return slowly to the desired track. It can be seen that although ship 3 maintains a similar track to ship 2, the heading changes by roughly sixteen degrees. Making corrections for the current also affects the speed made good, reducing it to around ten knots, so that ship 3 falls behind ship 2.

This can be seen to approximately agree with the mathematical solution of the example:

Ship's speed through the water 10.5 knots

Ship's desired track 090 degrees

Current drift 3 knots

Current set 000 degrees

Therefore

sin (Course Correction) = 3/10.5

Course Correction = 16.6 degrees

speed made good = square root (10.5 * 10.5 - 3 * 3)

speed made good = 10.06 knots

Of course, many other small factors make the calculation of the ship position much more complex, for example: the additional helm applied on ship 3 will slow it down; the distance the ship is initially pushed off track will make a greater course correction necessary; the auto pilot response and counter helm settings will affect how quickly the ship settles onto the corrected course; etc.

Repeating the exercise once more, one of the most obvious differences is the visual aspect of the target ship when viewed from the bridge of an observing vessel. In Figure 8-18 ship 0 has been placed at a position where it can observe ship 2 and ship 3.

Remembering that ship 2 and ship 3 are actually performing the same exercise, it is the difference between the developing situation with and without current that are being considered here. If the situation for each is taken in turn and consideration given to what action should be taken to avoid collision, had automatic avoidance been switched on. Then taking the situation between ship 0 and ship 2 and ignoring ship 3. Ship 2 is not affected by the current and can be seen approaching head on, therefore both ship 0 and ship 2 should have altered course to starboard. Now, ignoring ship 2, ship 3 is affected by the current and it can be seen that the navigator on ship 3 may have considered this a crossing situation, therefore, ship 0 should have altered course. Certainly as the current increases or the ship reduces speed and the navigator increases the course correction, this will become the case. This goes some way to demonstrating why, for realism, the target ships on a simulator should be affected by the current.

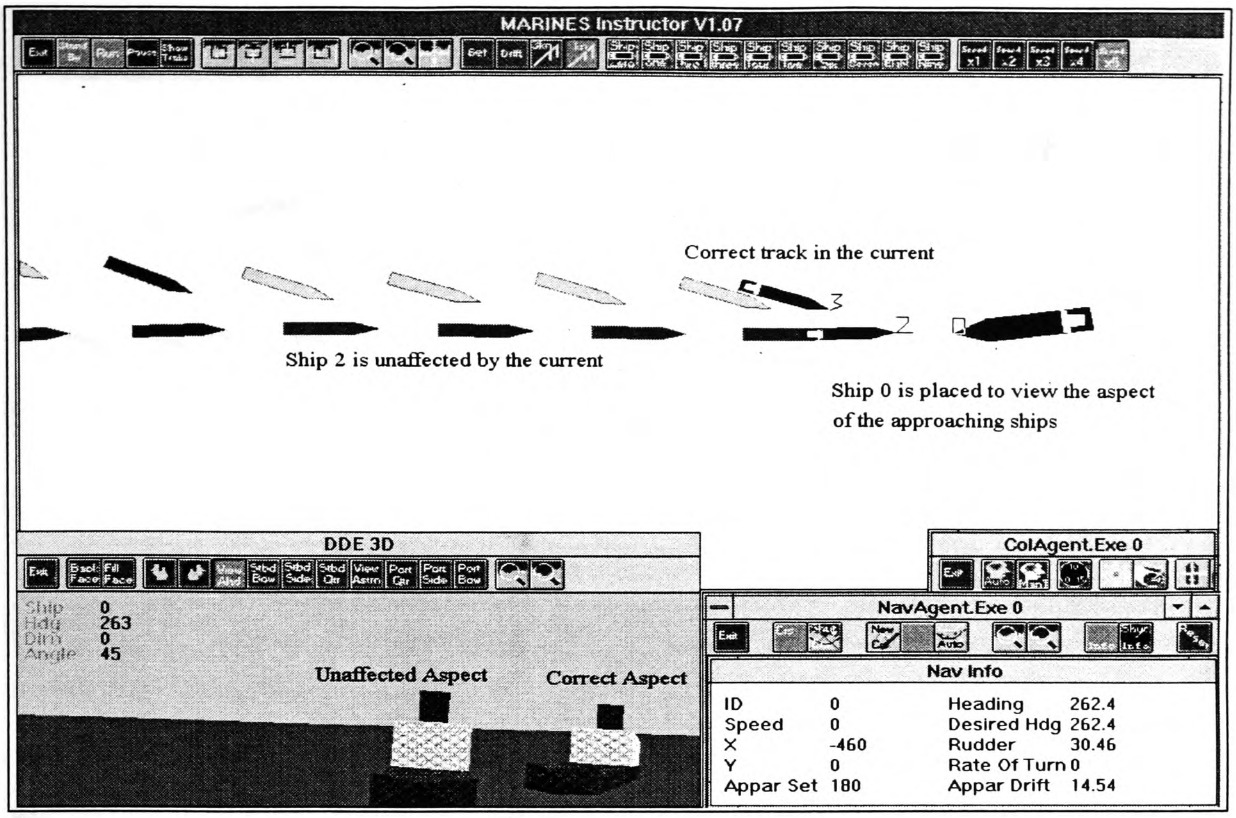


Figure 8-18 The difference in aspect of the ships can be clearly observed.

8.8.2 A changing collision situation for ships on reciprocal tracks when manoeuvring in a current

As described in section 0, the situation between two ships may change dynamically when the set and drift of the current change. This section demonstrates the one exercise performed under different current conditions. It shows how the agents in MARINES adapt to the different conditions at run-time.

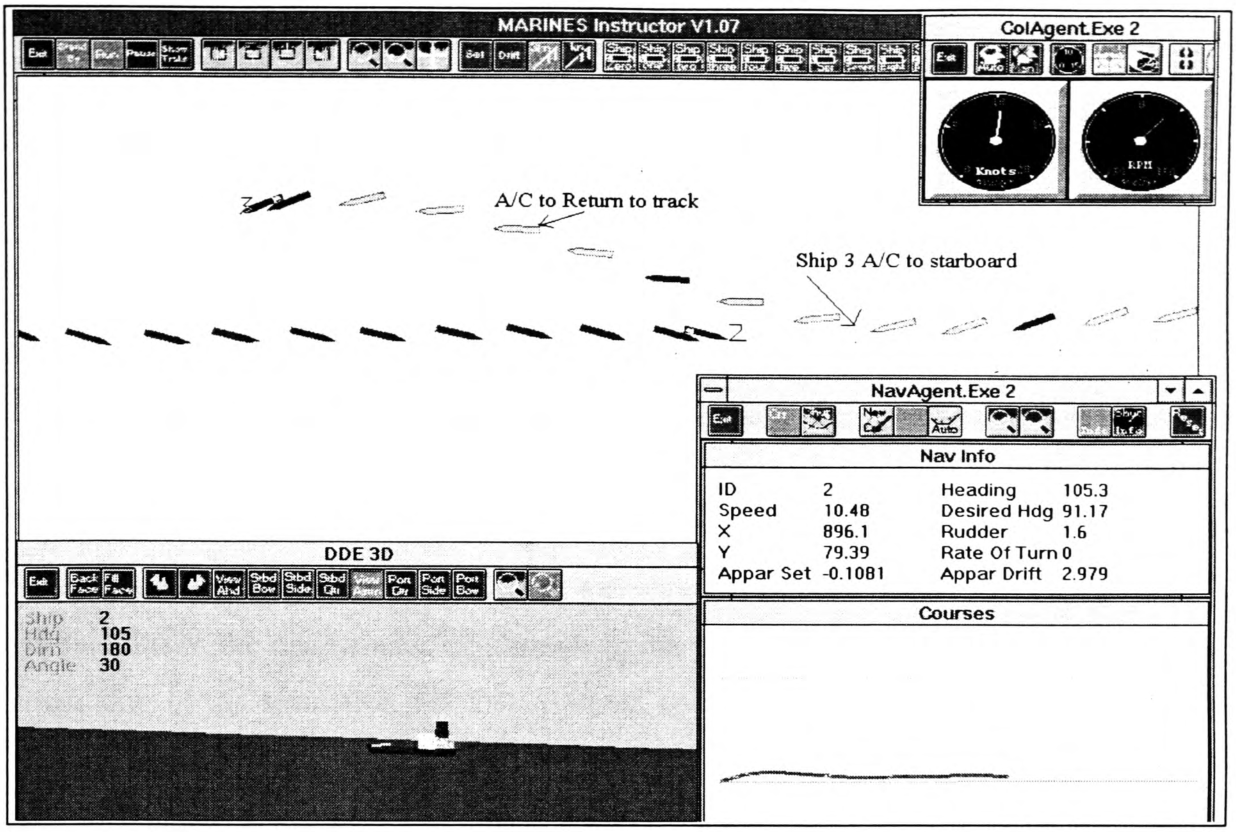


Figure 8-19 Reciprocal tracks with a northerly current

8.8.2.1 Reciprocal tracks

Once again, two target ships are considered and an identical ship model is used for both ship 2 and ship 3. In this exercise the track keeping agents are trying to make good reciprocal tracks. The collision avoidance agents are enabled and each use the same rule set. Each ship is running at approximately 10.5 knots.

In Figure 8-11 there is no current affecting the scenario and the two ships meet head on. As can be seen the ships both alter course to starboard, as would be expected.

In Figure 8-19 the exercise is repeated, however, in this case the ships are both affected by a northerly current of three knots. The situation has become a crossing situation under the rules and ship 3 has to give way. As can be seen this is what the collision avoidance agent does, altering course to starboard, as expected. The agent on ship 2 has a responsibility to stand-on and maintains course and speed.

The exercise was repeated once more, with both ships affected by a southerly current, although this is not shown as it is so similar. The responsibilities are reversed and the agents amend their behaviour accordingly. The agent on ship 2 altering course to starboard and the agent on ship 3 maintaining course and speed.

8.8.3 Reciprocal tracks, one ship unaffected by current

The exercise shown in Figure 8-11 was repeated once more. However, in this instance, ship 2 is unaffected by the current. The agent aboard ship 2 detects ship 3 as approaching head on, or nearly head on and alters course to starboard accordingly, as does the agent on ship 3. As shown in Figure 8-20 the resulting alterations are similar to the results for a head on situation. Whether, ship 2 should have altered course in the true situation is questionable. However, the radar plot gave the agent on ship 2 the wrong course and speed for ship 3; this was approximately 10 knots on a course of 270, in fact this is the course and speed made good. Had this been a student navigating the own ship of the simulator, the visual aspect of the ship and the ARPA vector would have produced conflicting information. It would have been a valuable lesson to watch for a change in compass bearing, which in this case would not have appreciably changed. The scenario more accurately portrays ship 3 as a high sided ship suffering considerable leeway, than two ships encountering the same set and drift of the current.

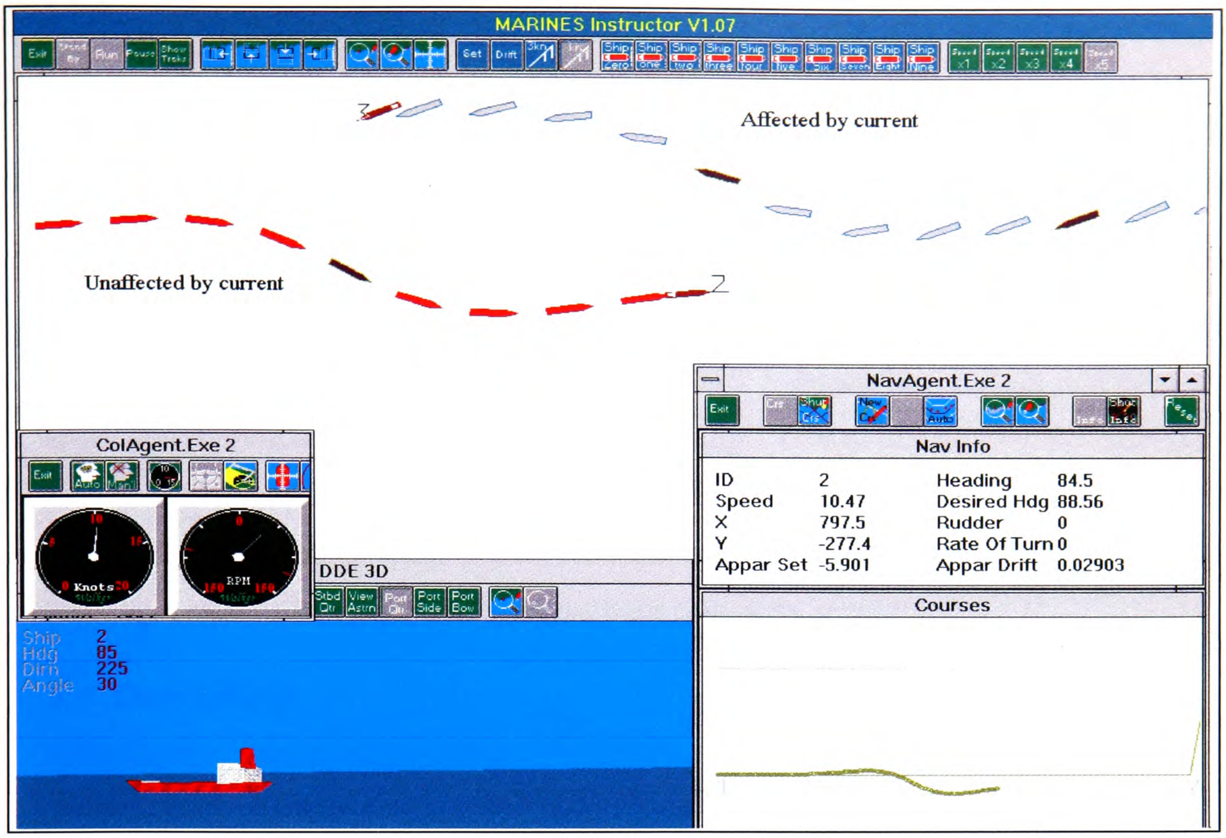


Figure 8-20 Reciprocal tracks, ship 2 is unaffected by the current.

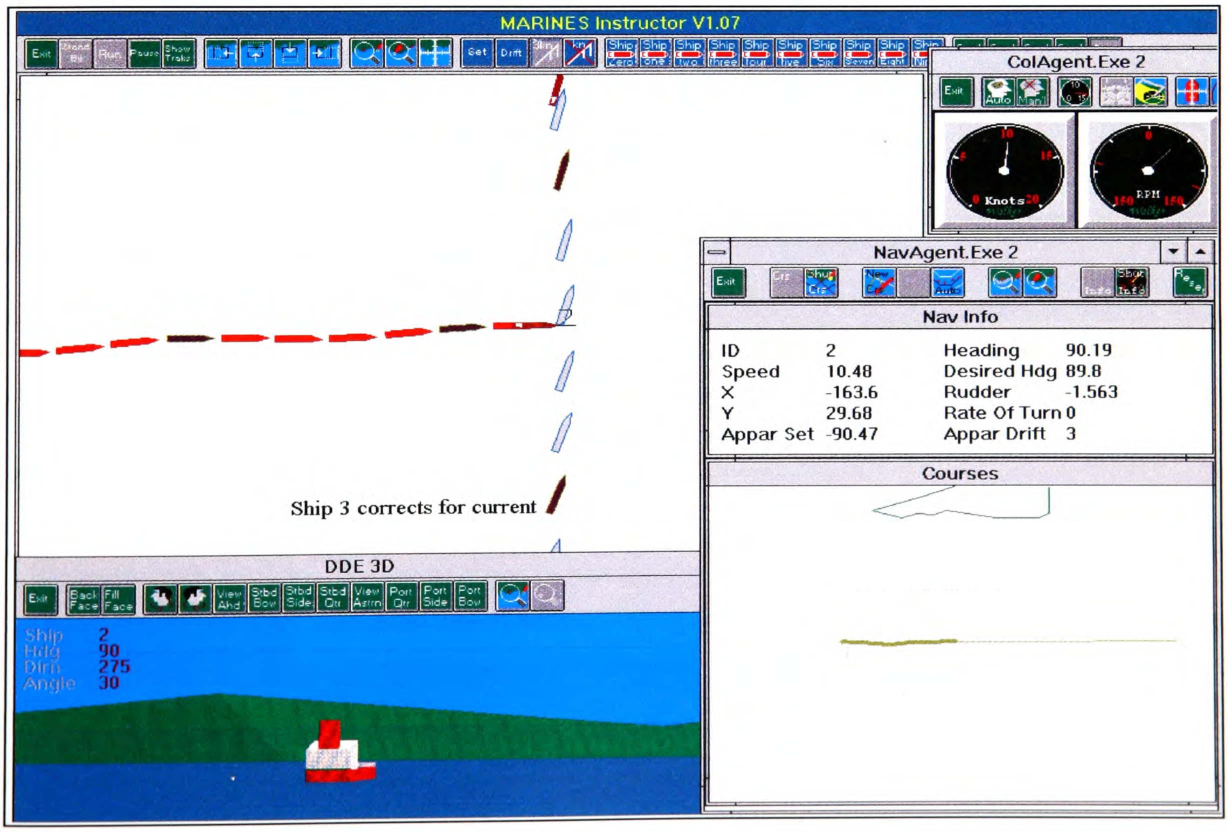


Figure 8-21 Crossing exercise with a strong westerly current.

8.8.4 Crossing situations

The crossing scenario is considered next. In Figure 8-12 with no current the collision avoidance agent on ship 2 detected a close quarters situation developing with a ship crossing from starboard to port.

The same exercise was repeated with a current setting 270(T) at 3.0 knots, as shown in Figure 8-21. Ship 3 is affected by the current and after initially moving off track to the west the track-keeping agent on ship 3 detects the current and begins to apply corrections to return to track. However, ship 2 is also affected by the current and only makes good approximately 7.5 knots. This means that the two ships do not actually have a close quarters encounter. The collision avoidance agents determine that no danger exists and dynamically adjust their decisions accordingly.

8.9 Overtaking

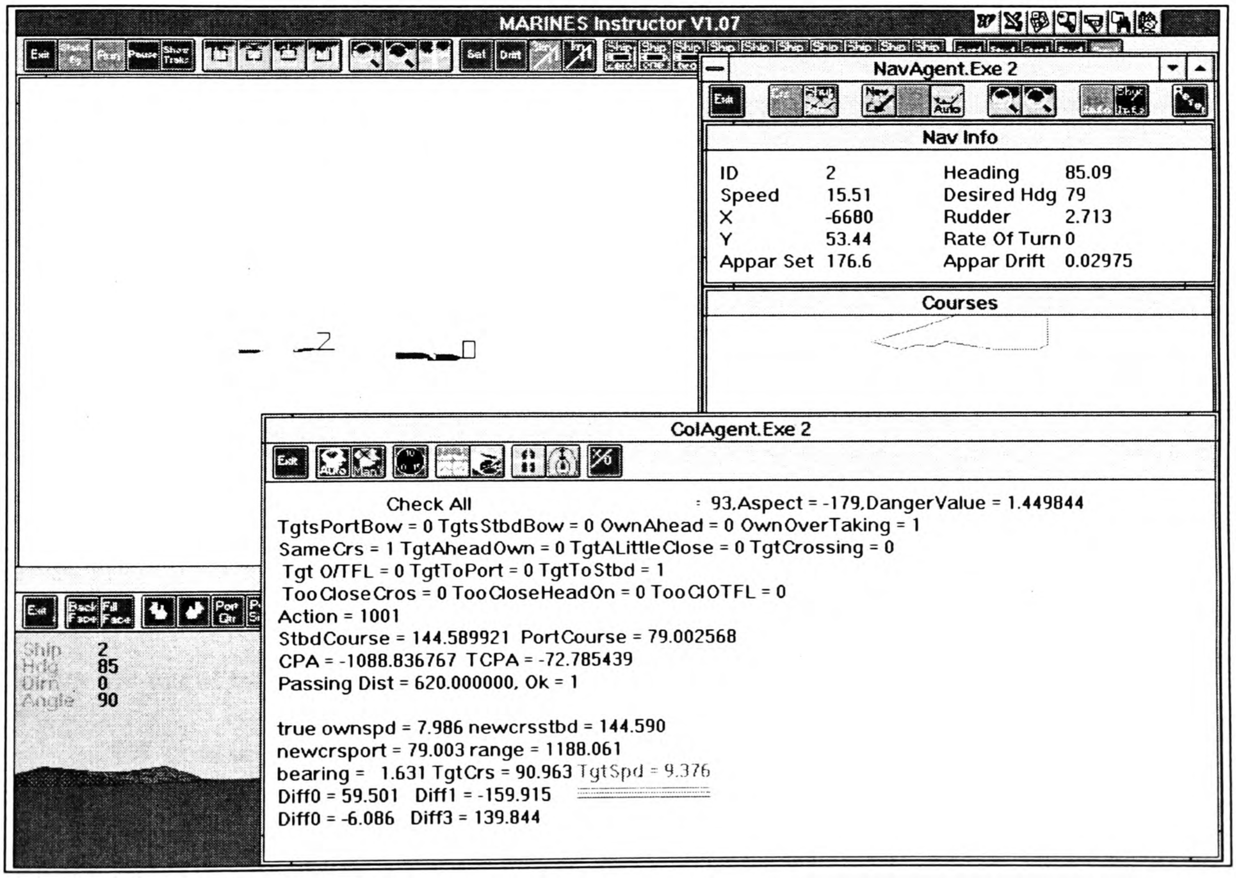


Figure 8-22 Overtaking without current

8.9.1 One ship affected by the current and the other not affected

The sequence of pictures Figure 8-1 and Figure 8-22 .. Figure 8-24 demonstrate the problem set out in chapter 4, when one ship is affected by the current and the other is not. Note that this is not normally the case in MARINES, the effect has been intentionally introduced to demonstrate the problem that occurs on many simulators. An exercise where ship 2 is overtaking ship 0 is repeated three times. In these examples ship 2 is affected by the current and ship 0 is unaffected by the current. In Figure 8-22 no drift is encountered and the collision avoidance agent on ship 2 detects a ship it is overtaking that is travelling at 9.376 knots. In Figure 8-23 the exercise is repeated with a current setting 270 degrees (-90) at 3.0 knots and the agent on ship 2 determines that the ship it is overtaking is travelling at 12.687 knots, because ship 0 is unaffected by the current. In Figure 8-24 the exercise is repeated with a current setting 090 degrees at 3.0 knots, this time the agent on ship 2 determines the ship travelling at 6.732 Knots.

The effect results in the agent determining a very different manoeuvre to avoid the target ship, than if both were affected by the current. Firstly, the calculated danger coefficient is different, due to the change in CPA and TCPA. This causes the agent on ship 2 to make the manoeuvre at different times. The agent alters later for the apparently faster ship, estimated at 12.687 knots, and makes an early alteration when overtaking the apparently slower ship estimated at 6.372 knots. Furthermore, the alteration of course is made from about 090(T) to 079 (T) for the situation without current and a speed differential of 6.134 knots. Despite the later alteration the agent on ship 2 only needs to make a very slight alteration to 089(T) for the apparently faster ship when the current is 270(T), the difference in speed made good only being around 2.8 knots. When the current is reversed to 090(T) then with a closing speed of 8.8 knots the agent on ship 2 has to alter to 054(T) to achieve the desired passing distance.

Of course, if the current were from the north or the south then the estimated course of ship 0 would be affected more than the estimated speed. This is shown in a similar exercise in Figure 8-25 where the speed of ship 0 is estimated at 10.33 knots by the agent on ship 2 and the course is estimated at 110(T). It can be seen that ship 0 is actually steering and making good around 090(T). The true speed of ship 0 is about 9.3 knots.

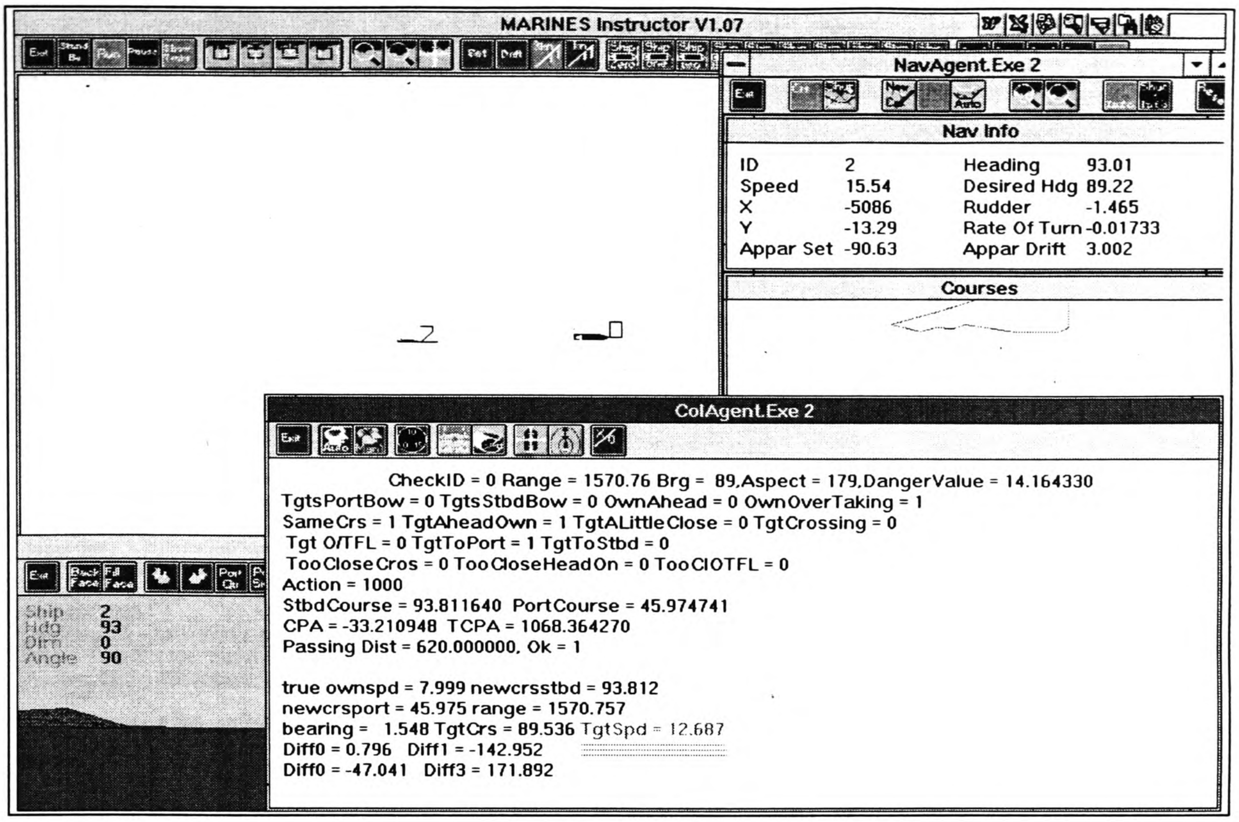


Figure 8-23 Overtaking with a three knot westerly current

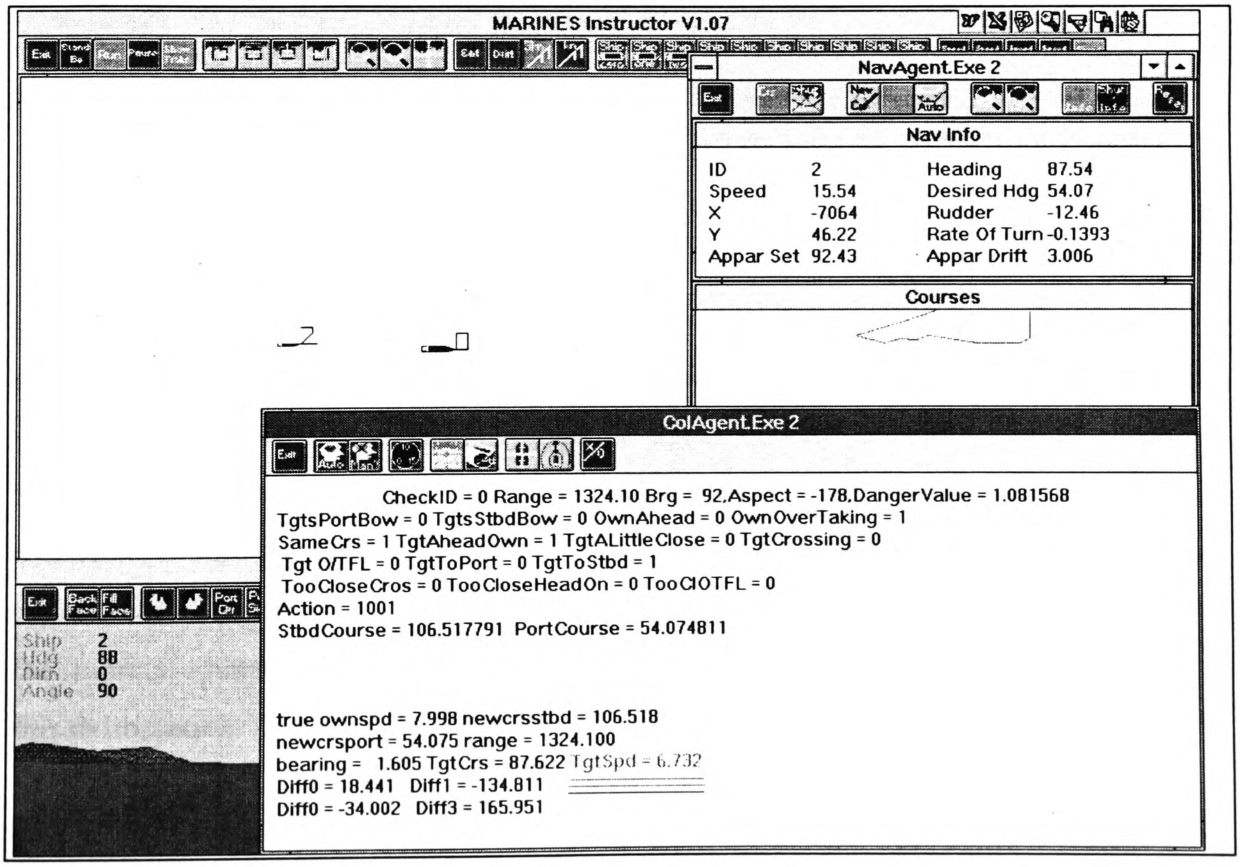


Figure 8-24 Overtaking with a three knot easterly current

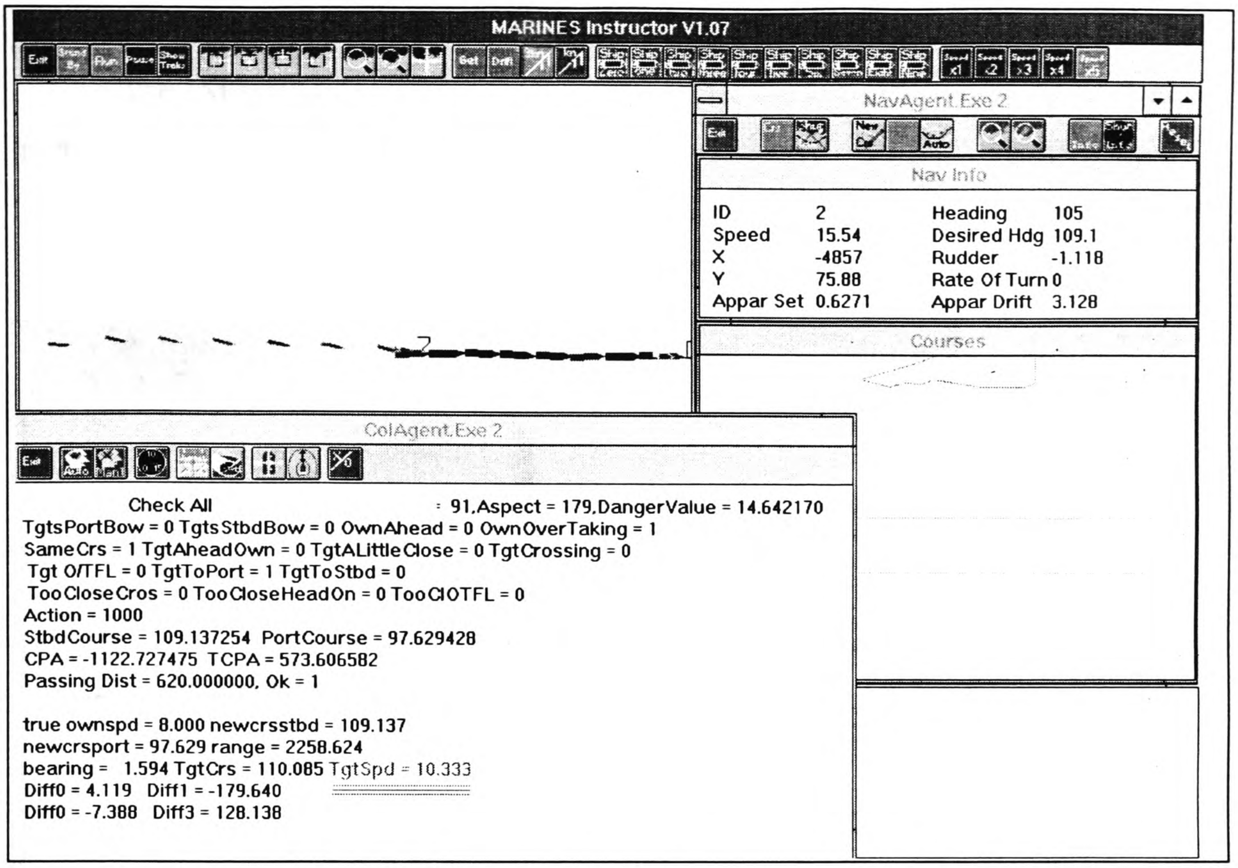


Figure 8-25 Overtaking with a three knot northerly current

8.10 Return to track

Choosing the correct time and course to steer to return to track is also important. The change of heading to return to track should occur once the agent has realised that it has successfully passed the danger. As can be seen in the examples depicted in Figure 8-9 to Figure 8-12, the agents successfully returned the ship to track after each deviation in the test exercises. In order to achieve this the track-keeping agent continually advises the collision avoidance agent of the desired course to steer, to maintain track. The collision avoidance agent adopts this suggested course whenever it considers it safe to do so. This depends upon the collision agent believing that the desired course to return to track will not infringe the domain of another vessel. Due to the imprecise nature of the problem, the agent also allows a safety margin when making this deliberation.

8.11 Increasing the number of ships

The MARINES system is restricted by incomplete collision avoidance, particularly for multi-target situations. The following tests have been devised to ascertain the strengths and weaknesses of MARINES when dealing with several target vessels. In addition, the tests should help to direct further research.

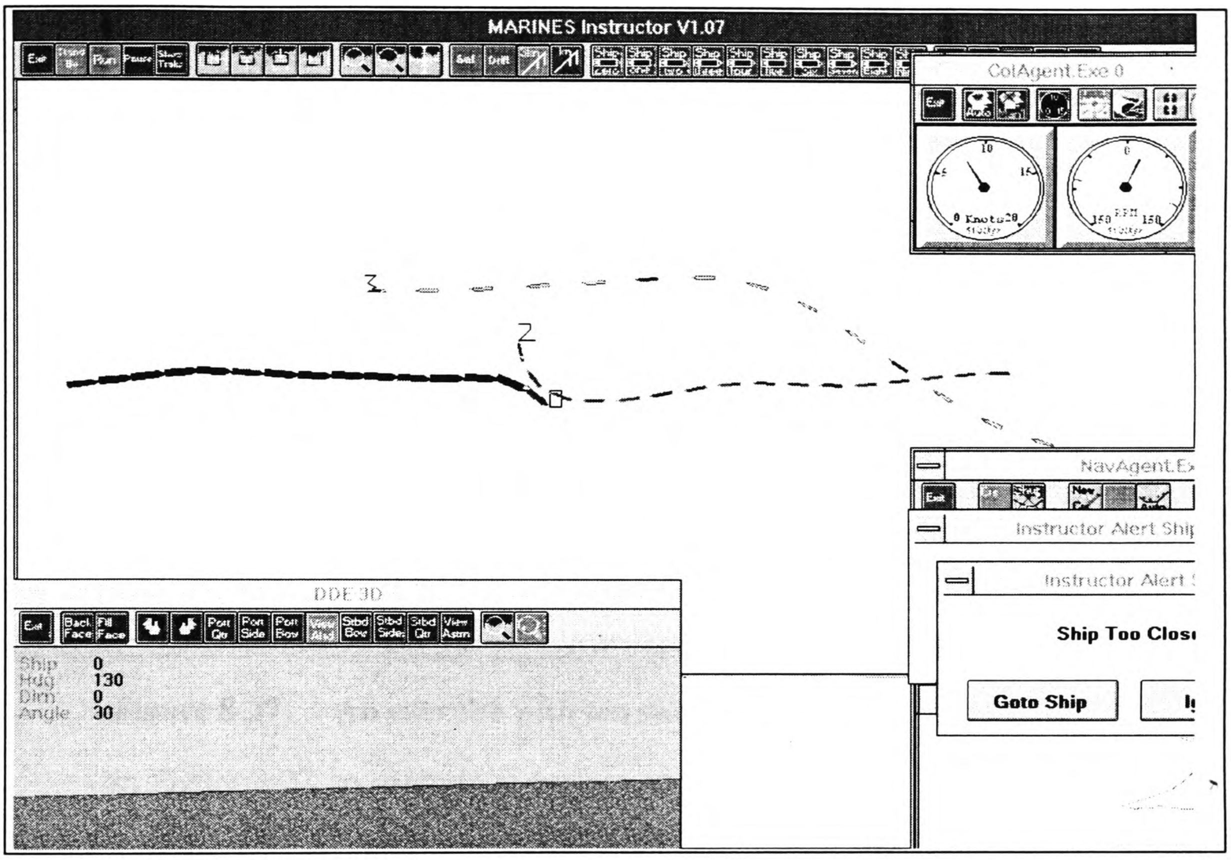


Figure 8-26 An exercise with three ships

In Figure 8-26 an exercise is run with three ships. It can be seen that the agent on ship 3 alters course to starboard to avoid ship 2 in an overtaking situation, also missing ship 0 in what would have been an almost head on situation. However, this manoeuvre confuses the situation between ship 0 and ship 2. The agent on ship 0 detects ship 3 as the most dangerous target, having a smaller TCPA. No action is required for ship 3 and none is taken. The agent on ship 2 is unable to alter course due to the close proximity of ship 3 overtaking. When ships 0 and 2 become too close both agents make a reactive manoeuvre to starboard. However, this is very late and results in a close quarters situation. There is obviously the need to tackle avoidance manoeuvres for multiple ship scenarios. More sophisticated planning is needed taking all the vessels into account.

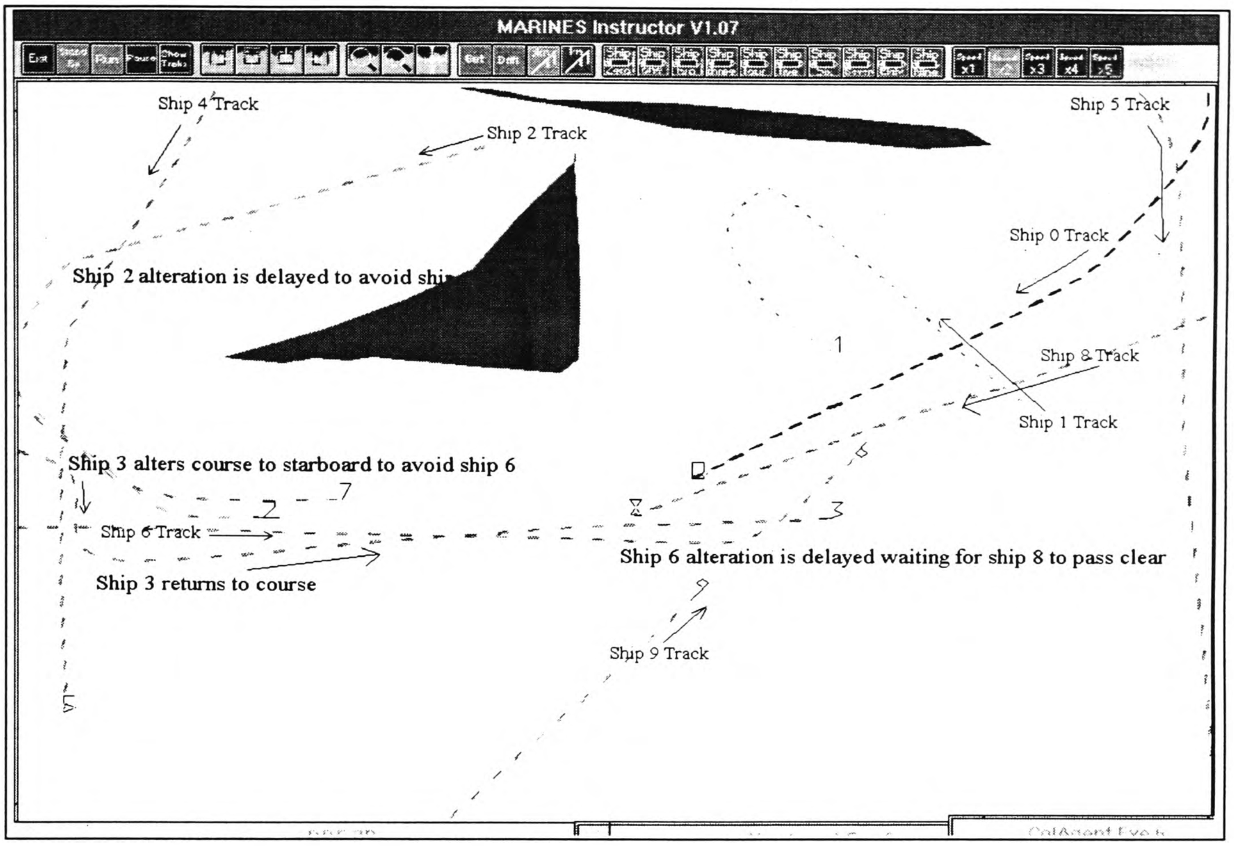


Figure 8-27 An exercise with ten ships

In Figure 8-27 an exercise is performed with ten ships. It can be seen that the agent on ship 3 alters course to avoid ship 6 which it is technically overtaking, although in fact both ships are making good almost the same speed. When the agent on ship 3 resumes course the alteration has delayed it so that no further action is needed. The agent on ship seven delays making an alteration onto a new course to avoid ship 2. The agent on ship 6 also delays a change to a new course to avoid ship eight.

The agent makes a check for dangerous situations before returning to course. This is determined by whether any future targets would infringe a desired CPA within a certain TCPA. Initially, a long desired TCPA meant that ships seldom returned to track when many ships were in the vicinity. A large number of off track warnings occurred making the instructor manually override several manoeuvres. The desired TCPA had to be reduced allowing the manoeuvring vessel to return to track sooner. It is believed that this should adjust dynamically according to the traffic density and danger, much as suggested with dynamic domains and arenas.

Additionally, the same exercise was also performed at a rate of five times world time. Loss of messages and delays prevented any manoeuvring actions similar to those shown in Figure 8-27. The ships came too close and again a number of warnings were issued. However, the author performing the experiment was unable to intervene quickly enough and in one case a collision occurred. Track-keeping agent 2 suffered an out of segment bounds faults, also causing a fault in collision avoidance agent 2. However, the instructor station and simulation continued running.

8.12 Architecture Exercises

8.12.1 Starting and stopping agents

The dynamic starting and stopping of agent processes at run-time may be desirable for several reasons. Firstly, different versions of processes can replace faulty ones without stopping the simulation. Secondly, if hardware resources are limited then the instructor is able to run the required agents as they are needed and discard agents that are no longer required. Thirdly, agents may be replaced by other agents with different characteristics.

Exercises were designed to determine if agent processes could be started and stopped dynamically. The initial tests were performed by starting the MARINES system without any agent processes attached, agents were then introduced and stopped dynamically.

8.12.2 Robustness

One of the problem areas identified by Chen (1992) was the fragile nature of simulator systems in general. Proponents of Distributed Artificial Intelligence (DAI) have claimed improved fault tolerance for DAI techniques in general and MASs in particular (Levy and Rosenschein 1991). The following tests are designed to examine the credibility of such claims with respect to MARINES.

8.12.2.1 Consistent performance

The agent processes and 3D process are designed to give priority to the main instructor process. This occurs in three ways. Firstly, the instructor process will selectively discard messages if too many requests for information arrive together; as many other processes are event driven by the messages arriving, this reduces their processing time. Secondly, the processes are designed to release the processor after a limited time-step, whether they have finished processing or not. Thirdly, the processes will disconnect themselves if no messages have arrived during a certain period; the 3D view will disconnect after the shortest delay, thereby reducing the overall system loading and allowing the other processes more processor time. This third method requires further research, it works well for slow machines but can be fooled into disconnecting if another process fails to act co-operatively.

8.12.2.2 Recovery from Floating Point Errors

In order to assist in the evaluation of the recovery from run-time errors, a function has been added to the collision avoidance agent to intentionally force a Floating Point Error (FPE).

The first scenario is designed to determine if a FPE in one process will affect another process connected by DDE as a client and server. A simple scenario was used with only one collision agent attached to the environment. The DDE_3D view was also connected, to detect whether any errors would cause further errors in other processes. A FPE was generated, terminating this agent application, then a further agent was launched. This was repeated for ten agent processes, running one at a time. This scenario was repeated three times, a total of thirty run-time errors, without causing any errors in either the instructor or the DDE_3D process. Unfortunately, the FP errors do cause the program manager window to be displayed in place of the MARINES environment, requiring instructor intervention. Nevertheless, the simulation continued without error.

In the next scenario all ten collision agents were attached to the environment, the DDE_3D view was also connected, however no track-keeping agents were attached. A FPE was generated, terminating each agent application in turn. This was repeated for all ten agent processes, until no agents were attached. Once again, this scenario was initially repeated three times, a total of thirty run-time errors. This did not cause any errors in the instructor environment, however, the DDE_3D process did suffer an internal FPE and terminate after several FPEs within the agents during each run. The experiment was therefore repeated six more times, each time the DDE_3D process terminated after either seven, eight or nine agent FPEs.

8.12.2.3 Re-configuration after loss of one agent

In MARINES the collision avoidance agent takes command and makes the final decision upon the ship manoeuvring. The track keeping agent acts as a junior officer and advises the collision avoidance agent. However, if the collision avoidance agent process fails then the track keeping agent is designed to take command of the ship. The next scenarios were intended to test the effect of loss of the collision avoidance agent upon the track keeping agent and evaluate whether the re-configuration operates correctly. In the first test one collision agent and one track-keeping agent were connected to the environment. A FPE was induced and the collision agent process was terminated, after 60 seconds two warnings were given by the track keeping agent which began to transmit messages directly to the instructor environment. The test was then repeated with ten collision avoidance agents and ten track keeping agents attached to the environment. The collision agents being stopped at intervals of five seconds. Eight of the track keeping agents reconfigured themselves to send manoeuvring instructions directly after approximately 60 seconds, the remaining two taking as much as five minutes to reconfigure. The agents issued two warnings each as they did so. Occasionally, the loss of a collision avoidance agent will bring down a navigational agent as well, but this did not occur during these tests.

8.13 Summary

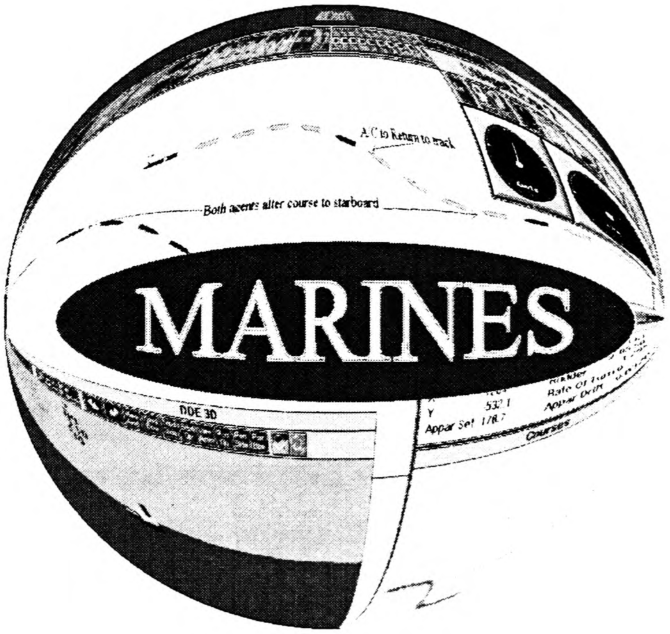
This chapter has reported on the more important results that have been obtained from running the MARINES test bed. Exercises have been performed that consider the individual components of the system; the manoeuvring characteristics of the ship models; the collision avoidance manoeuvres for head on, crossing and overtaking situations; and the track keeping characteristics of the different ships and *agents. Following this, exercises are performed with both collision avoidance and track keeping together.*

On many simulators the target ships are not affected by the current, and on many simulators that have sea stabilised target ships, the instructor is left with the task of correcting for changes of set and drift. In order to emulate these effects, MARINES includes the ability to switch on and off the current for a specific target ship. Automatic correction for a changing current can also be provided. Experiments have been repeated to permit a comparison between ship motion and interaction with these features enabled and disabled in various situations. The change of aspect of approaching target ships has also been demonstrated, for land stabilised and sea stabilised ships. After a manoeuvre to avoid another ship, return to track is achieved reasonably quickly without unduly large course corrections.

Finally, exercises were performed to determine the value of the robust nature of a Multi-Agent Systems approach to the marine instructor station. These included recovery from floating point errors and reconfiguration after the loss of one agent.

The results obtained are also discussed in the next chapter in combination with the results of the user evaluation.

Chapter Nine



User Evaluation

9. User Evaluation

9.1 Overview

In parallel with experimental evaluation under controlled conditions, evaluation of the system was performed with a number of experienced simulator designers and users. This evaluation was performed over a period of time with some development taking place between each interview. This phased evaluation permitted a few key features to be introduced as they were suggested, rather than putting them forward as future work.

The overall results of the experiments performed in chapter eight and the user evaluation at the beginning of this chapter are discussed in section 9.3. This discussion is divided into several sections: ship model performance; track keeping performance; collision avoidance performance; performance when encountering the set and drift of a current. After this, in section 9.4, the use of the MARINES technology within a commercial instructor station is discussed.

9.2 Peer evaluation

The opinions of several experienced simulator operators and developers were sought throughout the MARINES project. I am indepted to all of these for their assistance.

Nielsen (1993) discusses different evaluation techniques for usability. Many of the techniques are more relevant to a complete commercial system. However, three areas were considered important when evaluating the MARINES system. The use of questionnaires, interviews and the prototyping of the system.

It has been recognised that questions on forms often receive answers that respondents think that they are expected to give, rather than what they truly believe. Additionally, response time for postal questionnaires can be lengthy and many people do not respond to postal questionnaires at all. Interviews tend to obtain far higher response rates and this was considered essential for the MARINES project, due to the small number of experts available. A semi-structured interview technique was adopted, and a number of exercises were worked through with the interviewees to raise the central issues. This technique has the advantage of being reasonably free form; follow up questions can be introduced that have more relevance to the direction the interview is taking. Additionally, in interviews of this type questions can be rephrased if it becomes clear that the respondent does not understand. Therefore this technique should lead to a more in-depth discussion, more accurate expression of views and immediate results. The disadvantage that Nielsen identifies is the lack of easily comparable results. Therefore, in this case, the interviewees were also asked to fill in a questionnaire after the demonstrations and discussion. As the respondents had already recorded their views on tape and had had time to form an opinion, it is hoped that the results of the questionnaires were reasonably accurate.

In line with the idea of prototyping, also recommended by Nielsen, the interviews were conducted over a period of time. Solutions to some of the stronger recommendations that came from the interviews were implemented before the next interview took place. This provided a continually improving product, although the underlying functionality remained the same. In order to keep the number of programming alterations to a minimum, changes were generally only made where the respondent was able to rationalise the need for the change with concrete evidence. For example, evidence such as, a ship's rate of turn being too slow supported by actual figures from a ship's performance trial was considered incontrovertible and the change was made. A feeling that a desired feature would be an improvement was not considered sufficient reason unless several respondents made the same observation, although if an observation was made it was intentionally raised with later interviewees so that important observations were not overlooked. The disadvantage of these changes, once again, is the lack of fully comparable results. It is difficult to find an even platform to compare the results of the earliest tests with the later ones. Also, at one stage the changes actually led to the temporary inclusion of an error, reducing the systems effectiveness. However, after the error had been eliminated the tests were run again with the same interviewee. In order to reduce the discrepancies between the earlier and later tests a few of the more controversial exercises were re-run after the changes. The results were printed and the views of some of the earlier respondents were sought about these later results, to see if the changes affected their earlier choices.

The survey results support the experimental findings, discussed in the last chapter. This close correlation between the expert user opinions and the experimental results is considered a positive endorsement of the direction the research has taken. This is to some extent due to the phased evaluation that has taken place.

9.2.1 Starting out

From the outset a number of experienced master mariners, simulator instructors and simulator developers took part in valuable discussions about the concept of MARINES from a marine simulation viewpoint. These discussions helped to clarify the direction of the research from an early stage. This also helped to identify ambiguities and omissions in the original project description.

The combined research group meeting at Warsash provided further ideas and also gave an early insight into existing research in similar directions and areas of special interest. This meeting also served to introduce the author to the developers of a collision avoidance system at Liverpool John Moores University. This team agreed to assist in the evaluation of the first prototype of MARINES. One of the weaknesses detected in MARINES v1.0 was the inability to decide when to commence an alteration. Action was usually taken too early, as soon as sufficient information existed to make a decision. A solution was found by adapting the danger coefficient of Smeaton and Coenen (1990). This danger coefficient goes some way to providing a dynamically varying Arena[1]. The use of this coefficient has been discussed in chapter eight.

In addition to the theoretical papers, a critique of the project objectives from a Multi-Agent System (MAS) perspective was obtained from two well known agent researchers. This helped to clarify the agent theories and architectures and, in particular, it was suggested that the agents should communicate directly and some architectural improvements were discussed. At this time only the collision avoidance agents had been implemented. Therefore, only one agent was attached to each ship and there was no direct inter-agent communication, all communications being between an agent and the environment.

[1] See chapter 2

The managing director of a simulator manufacturing company discussed the important aims and objectives. He is also a consultant in mathematical ship modelling, port design and simulation. Improvements were suggested for the motion of the ship models; the early models in MARINES were all of one ship type and had not been tuned completely. New ship models were introduced and some tuning took place, although most of the testing took place just before the main user evaluation.

The navigation and hydrography department of Plymouth University, have supervised many marine research projects. A collision avoidance system (Blackwell et al. 1989) was developed at Plymouth and used as the basis of PC Maritime's collision avoidance simulator. The lecturers and researchers were helpful in discussing the MARINES system from a marine viewpoint. At the time of the meeting only the collision avoidance agents had been implemented and the addition of track keeping was discussed with a proponent of these systems for integrated bridges. The realism and usability of the existing simulator at Plymouth was also discussed together with their plans and reasons for replacing the simulator. They also described the forthcoming improvements to the target ship motion promised by Transas Marine Ltd., who were to be the probable supplier of the new system. This target ship motion includes track keeping motion that allows for the set and drift of the current[2], although automatic collision avoidance is not supported.

9.2.2 Developing MARINES for track keeping

In order to investigate the way in which ships react to changes in environmental conditions two additional features were called for. The first was a track keeping agent. This had been identified as a desirable option, and the discussions in Plymouth had also supported the need for track keeping that was more realistic; replacing simple systems that just maintain a direct route. To demonstrate the more natural track keeping the second feature was necessary, the ships needed to be affected by the set and drift of the current.

---

[2] Transas Marine has recently released a simulator with more realistic track keeping. At the time of writing the Transas company has been kind enough to supply a few details. Up to 200 target ships are supported and are automatically guided along the track by the program, allowing for the set and drift of the current.

9.2.3 System evaluation and tuning

A questionnaire was then developed to elicit information from seven experienced simulator developers and instructors; a copy of the questionnaire is given in Appendix C. The first few questions were to elicit the expertise of the users and it can be seen in Figure 9-1 that the sample interviewed have a wealth of expertise in the simulation field; knowledgeable, not only in using simulators for training but in their design and use in port design and in developing ship models. The majority also have seagoing experience and are well versed in the collision avoidance regulations.

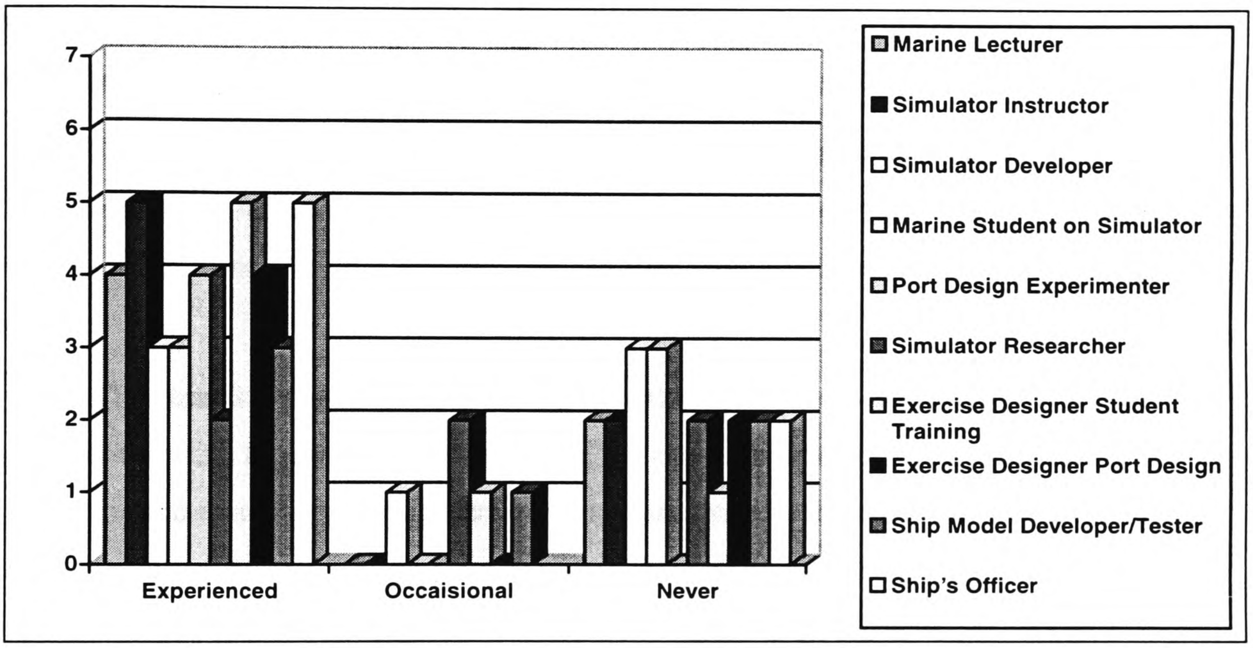


Figure 9-1 Marine Simulator Experience

9.2.3.1 A description of the user exercises

This section provides an overview of the exercises performed with the users. The discussion also highlights specific improvements and features suggested by specific interviewees and some changes that were made to address these suggestions.

The user evaluation tests of the completed MARINES systems began with a software engineer who had designed several marine simulator systems. Prior to this, the interviewee had also acted as an instructor at **CA**rdiff **S**hip **SIM**ulator (CASSIM). Various exercises were performed including head-on, crossing and overtaking manoeuvres. The interviewee was familiar with the MARINES system, several features having already been discussed during the development process.

Similar experiments were then performed with two simulator developers and users at Maritime Dynamics Ltd. Several additional improvements were suggested: the inclusion of historical tracks for all the ships in the scenario was considered essential, because at this stage only one track history was available, and comparison was difficult; the roll period of all the ships and particularly the super tanker needed to be increased; the inclusion of a simulation rate control was desirable, at the time of the experiments the simulation only ran at one rate, although this could be varied at compile time. Features were duly implemented in response to these suggestions.

The next interviewee runs a small but growing simulation and training centre in the Philippines. The centre uses a Transas simulator which they have developed from a small, simple, single ship simulator to a fully functional system over a number of years.

The results were somewhat different in this interview; in general the respondent believes that the target vessels are there to cause the student to perform an action. Therefore, the instructor must have complete and precise control over the routes of the target vessels. The target vessels must go where they are designed to travel by the instructor, as is the case in most simulators. The results were considered acceptable but the value of such a system for basic exercises was doubted.

However, it was considered that the lack of realistic target ship motion due to the set and drift of the current was a failing of the present generation of simulators. In this respect and for exercises involving a large number of target ships, the provision of collision avoidance and track keeping was considered desirable. Although there was doubt that the target ships should manoeuvre for the students' own ship; it being necessary to force the own ship team to react for the rogue vessel. The manoeuvres were considered sufficiently realistic for the exercises that were performed. The suggestion to use the 'aspect'[3] of an approaching vessel for the manoeuvres was taken

[3] The visual aspect of an approaching ship is used to determine which ship should give way under the collision regulations. See Appendix A for more details.

up at this point, assisting the radar detection. Unfortunately, as will be described, this introduced an error into the system that was discovered in the next evaluation where the system was demonstrated.

The next meeting was in Gothenburg, Sweden, the concepts and results of the simulation exercises were discussed. Several users were at this meeting but the system was not demonstrated as a suitable computer was not available. Therefore the views have been valued as a single evaluation and only one interview was recorded. The discussion was relevant as instructors at the centre had used PC Maritime's "Officer of the Watch" desktop simulator which has automatic collision avoidance between target ships. However, one of the respondents, who has used the system, observed that the set and drift of the current is not modelled on the PC Maritime simulator. As mentioned before the PC Maritime collision avoidance system is based upon work by Blackwell (Blackwell et al. 1988); Blackwell and Stockel (1989, 1990); and Blackwell et al. (1991) using domains. In use, this simulator is said to provide a very good standard of collision avoidance between the target ships. Some targets may be set as rogue vessels so that they do not avoid other vessels. However, as set and drift are not modelled, the only dynamic changes to the exercises are instigated by the manoeuvring of the own ship. Therefore, the exercises may be designed and tested before the students use the system and an instructor does not need to be in attendance at run time. There is little doubt that the simulation is highly advanced for a commercial product that has been on the market for some time.

The general feeling at Gothenburg was that as the remaining elements of micro based simulators improve the target ships will need to have more realistic motion. In the instructors' experience, most of the existing simulators fail to provide all of the fundamental requirements. If all the best parts of the existing simulators were combined into a single simulator then more realistic target ship motion would be highly desirable. From the limited results of the MARINES screen dumps that were assessed the instructors agreed that the manoeuvres were acceptable and the accurate target ship motion was an important step forward for simulators.

The next interviewee has spent many years as the master of a tanker and undergone training on several marine simulators. This proved to be one of the most intensive evaluations; a wide variety of exercises were performed extending for several hours. This Captain was particularly interested in creating exercises that highlighted mistakes that he had seen people make at sea. On the one or two occasions when the agents did not perform as expected he recognised traits that inexperienced officers had shown. However, the results did uncover one error in the software. When an agent detected a ship that was overtaking it would sometimes try to alter course to avoid the overtaking ship. Provided that the overtaking ship made a suitable manoeuvre then after a few degrees alteration the ship being overtaken would return to course. Unfortunately, repeating the experiment several times failed to reproduce the error and although the code was inspected the error was not eliminated immediately.

As a lecturer in maritime studies and a simulator instructor at CArdiff Ship SIMulator (CASSIM) the next interviewee has considerable experience of marine simulators. He has used simulators for student training, port design and testing ship models. As an instructor at CASSIM he has controlled exercises containing up to ten target ships; as many as three of these could be tugs. As he explained, not everyone could produce realistic effects; an inexperienced instructor would often cause tugs to collide with the own ship or fail to get close enough to attach the ship's lines, and thus the target ships' realism could vary according to the ability of the instructor. In the initial test of the system the same software error found in the previous interview was exhibited again, this time an exercise was created that reproduced the error reliably, the speed and positioning of the ships involved being critical. Once this had been achieved it was relatively simple to eliminate the error. A combination of an error during re-initialisation of historical data for the second and subsequent exercises and an error calculating the aspect of a target ship in one segment proved to be the problem. This led to a fact sometimes being asserted erroneously within the inference engine and, infrequently, the agent would make the wrong decision based upon this belief. Once this error was eliminated the collision avoidance between two ships was found to be very realistic for head-on and overtaking manoeuvres. The error had been introduced after an earlier evaluation, prior to this the aspect of the ship had not been used. It was felt that action should be taken earlier for crossing situations. This can be achieved by increasing the danger coefficient, however, the other manoeuvres would also occur earlier. The ability to respond to the dynamically changing current was seen as an important factor. In answer to another suggestion the compass bearing of the most dangerous target was made available on a debug screen at run time, when action was imminently required. This assisted in the evaluation of the approaching target. In the future it would be valuable to permit the instructor to take compass bearings of approaching vessels using some form of cursor, possibly the mouse.

9.2.3.2 Expert user results

Due to the small number of seven expert users the graphs are only indicative of the trends that might be obtained. However, it was considered essential that the users were conversant with both the marine regulations and conventional marine simulators. Views were sought from one or two persons in other fields but most felt unable to comment upon many sections of the questionnaire. Therefore, only the views of the experienced users have been included in the results below.

- Ship models

The target ships on some simulators are navigated at run time by the instructor and some of these use realistic models. However, the onus, on such systems, is upon the instructor to keep all the ships headed in the correct direction; usually only a limited number of such ships are used at one time. In MARINES the models approximate to generic ship types and this was the answer that five of the seven respondents chose for computer generated target ships in a training simulator(section 2 question 1). Of the others, one of the respondents chose simplicity and the other realistic ships. After final tuning for the exercises in the previous chapter, the ship models were considered adequate for target ships, and a considerable improvement over many systems.

- Track keeping

Although not a specific question on the questionnaire, in conversation the track keeping was found to be reasonably realistic for all the target ship types. The different tracks for the different models, different rates of turn and general motion was considered to be adequate. The correct sea stabilised target speed and aspect for scenarios involving the set and drift of the current was considered important by all the interviewees. However, two of the respondents preferred the target ships to follow the pre-set routes precisely, without deviation, believing the instructor should intervene to alter course if necessary. One of these welcomed warning messages to alert the instructor to impending danger.

- Collision avoidance

Section 3 of the questionnaire covers the users opinions of the manoeuvres made by the ships for specific situations. This section took the longest to complete and some exercises were repeated for discussion while this section was filled in. The majority of the respondents were completely familiar with collision avoidance problems and felt confident about their assessment of the results; the instructor of a simulator has to discuss the students' manoeuvres in detail, suggesting improvements and explaining errors. Two users believed that the instructor should deal with the manoeuvres manually; however, one of these found all the manoeuvres acceptable and the other was satisfied with all the manoeuvres apart from crossing manoeuvres.

After the errors discovered had been tackled and the rules had been tuned to provide a solution to most collision scenarios involving two ships a series of experiments were re-run, trying to re-create the most demanding scenarios met in the user evaluation. Of course, the MARINES system is a complex continuous simulation environment that is undergoing continuous dynamic change. As with any such simulation, it is not possible to precisely replicate the exercises. Some of the expert users were approached again to analyse the results obtained, ensuring that they improved upon the previous exercises. The results obtained from the expert users are shown in Figure 9-2 .. Figure 9-8.

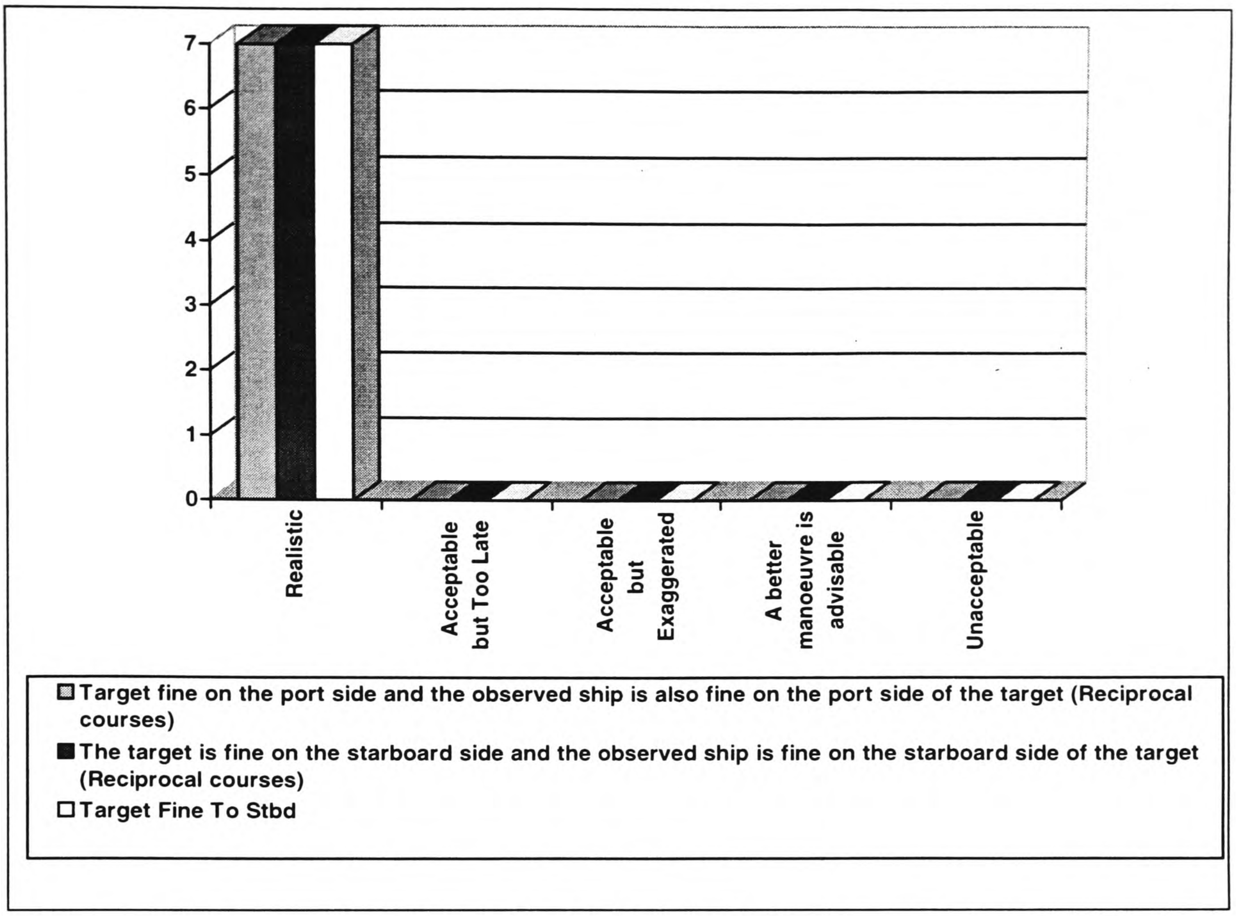


Figure 9-2 Head on without current affecting the ships

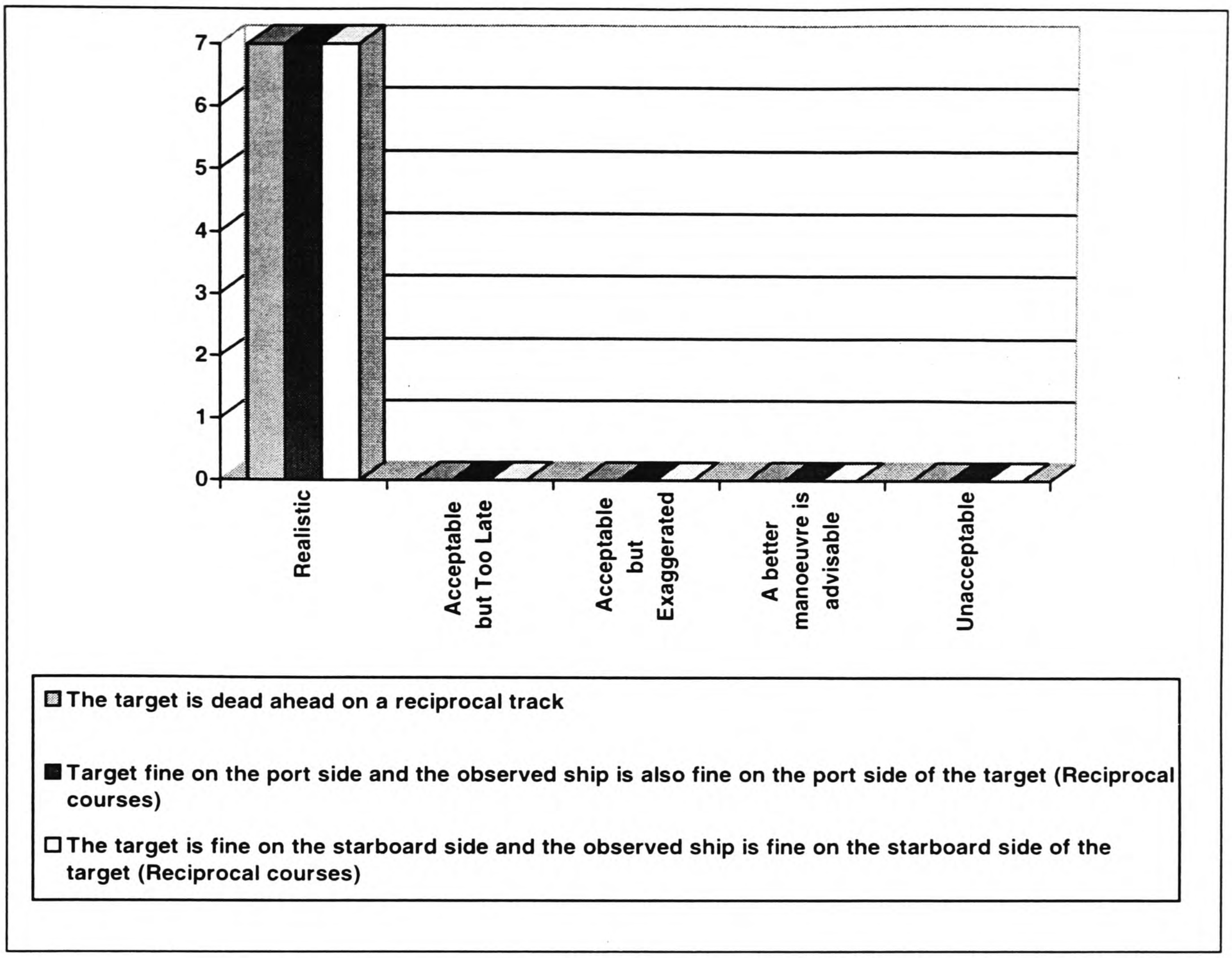


Figure 9-3 Head on with current and affecting both the ships

In head on scenarios involving only two ships, and without a current, all of the expert users agreed that the manoeuvres were realistic, the results are shown in Figure 9-2. This was encouraging and was reinforced when the set and drift of the current was introduced, again all of the users agreed that the manoeuvres were realistic, as shown in Figure 9-3. Note, that in this case some of the head on manoeuvres became crossing manoeuvres due adjustments of course to compensate for the current. This dynamic adjustment to the changing scenario was one of the objectives of the MARINES research.

The crossing scenarios were less generally accepted. The main criticism was the late manoeuvre for a ship that fails to give way when crossing from port to starboard. This was considered unacceptable by four of the users and too late, but acceptable, by three of the users. The reactive manoeuvre was too late and reflects upon a problem with the reactive component of the MARINES architecture. If the reactive component of the collision avoidance agent raises the alarm too early then the instructor is needlessly bothered by the alarm. However, if the alarm is left until the last possible moment then the instructor may be unable to make a manoeuvre rapidly enough to avoid collision. The problem is also seen at sea; determining the development of a dangerous situation early enough to summon the captain and make him fully aware of the situation is always difficult. Inexperienced officers may call the captain for situations that are not dangerous and fail to raise the alarm early enough for the truly dangerous situations. The original choice to raise the alarm depending only on the range of the target has not been successful. Several other algorithms have been tried, all of these have resulted in spurious warnings or, worse, failure to report. The navigator aboard a real ship will base this decision upon a number of factors including: how much history is available for the ship; if it is known, how close the ship is passing and how long before this occurs; the relative bearing and aspect of the ship; how much the compass bearing of the ship is changing; which ship is the give way vessel under the rules; and the type of ship involved. A decision based upon this amount of information is more than simple reaction. The architecture of the collision avoidance agents may need to be adapted to take these problems into account. This is considered in chapter eleven on future work.

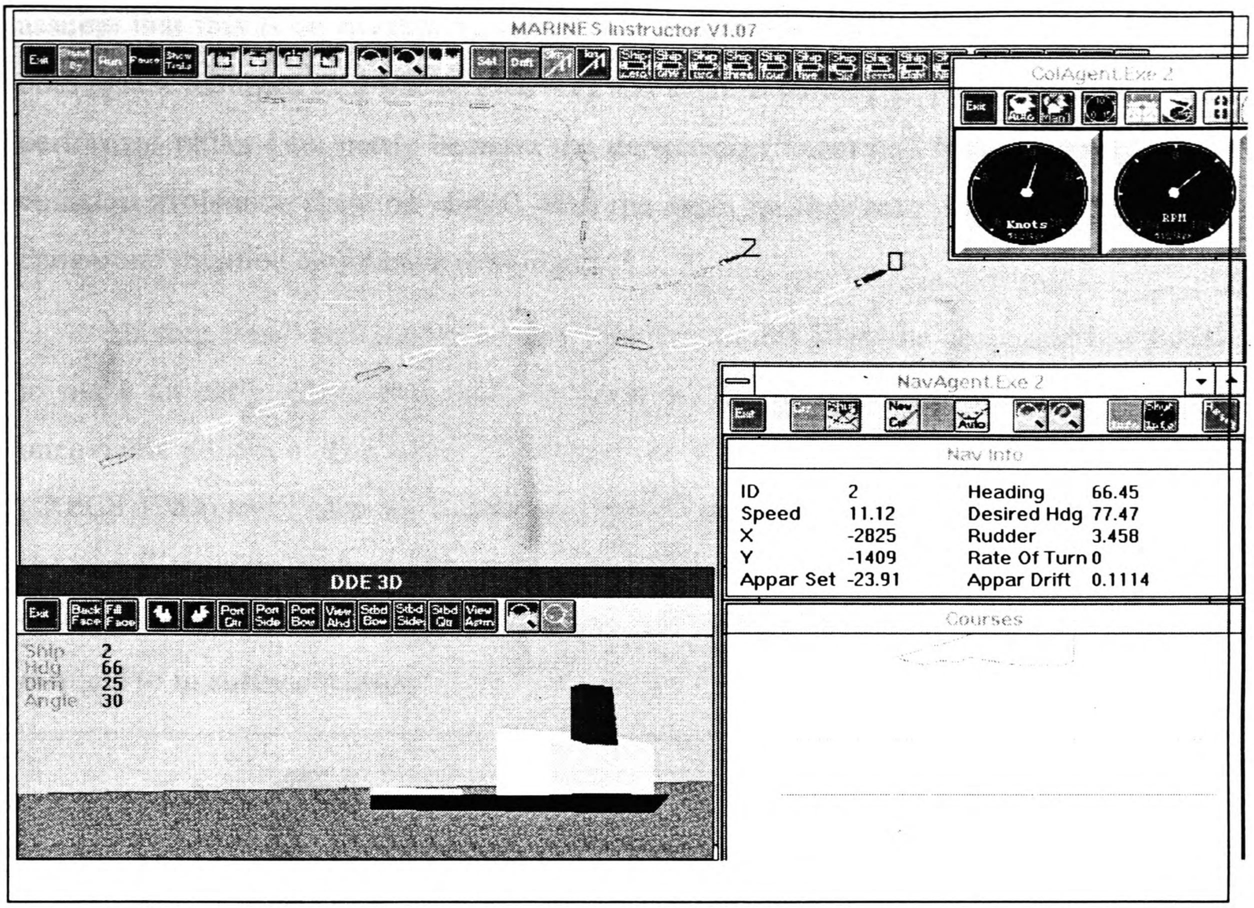


Figure 9-4 Ambiguous responsibility between two vessels leads to unnecessary manoeuvring

Collision avoidance for a ship approaching from two points[4] abaft the starboard beam[5] can be a difficult manoeuvre for a navigator to be faced with. In Figure 9-4 ship 2 is a give way vessel under the collision regulations. In this case the navigator aboard ship 2 observing ship 0 approaching from abaft the beam can make an accurate assessment of whether ship 0 is an overtaking or a crossing ship from the relative bearing of ship 0 when it is first detected. However, the navigator aboard the approaching ship 0 is unable to make an accurate judgement of the aspect and should

[4] Navigators commonly refer to a visual bearing in points relative to the heading, or some other fixed position, of the ship. It is more convenient and less open to misinterpretation to say "two points on the port bow" than "a relative bearing of about 337.5 degrees". The system refers to points of the compass, there are 32 points in 360 degrees. I.e. one point is 11.25 degrees.

[5] The beam, in this context is an imaginary line running 90/270 degrees to the heading line. "Forward of the beam" and "Abaft the beam" are terms to describe in general terms whether a ship or another object is ahead or behind the navigator's own ship. A ship that is first observed more than two points abaft the beam is an overtaking vessel according to the collision regulations (IRPCS 1989).

assume that this is an overtaking vessel and should therefore give way. In Figure 9-4 both agents assumed they had to give way and acted accordingly. The manoeuvres were performed rather late; partly because the danger coefficient had been reduced for the collision avoidance agent on ship 0. Had the agent on ship zero turned to port then a dangerous situation could have developed.

If ship 0 had actually been more than two points abaft the beam and had failed to make an early manoeuvre then the agent on the stand on ship 2 would be in an unenviable situation. The navigator aboard the stand on vessel should follow rule 17 (IRPCS 1989) and *"keep her course and speed"* until *"it becomes apparent to her that the vessel required to keep out of the way is not taking appropriate action"*. It is this situation and the mirror image for the port side that the agents in MARINES failed to respond to in sufficient time.

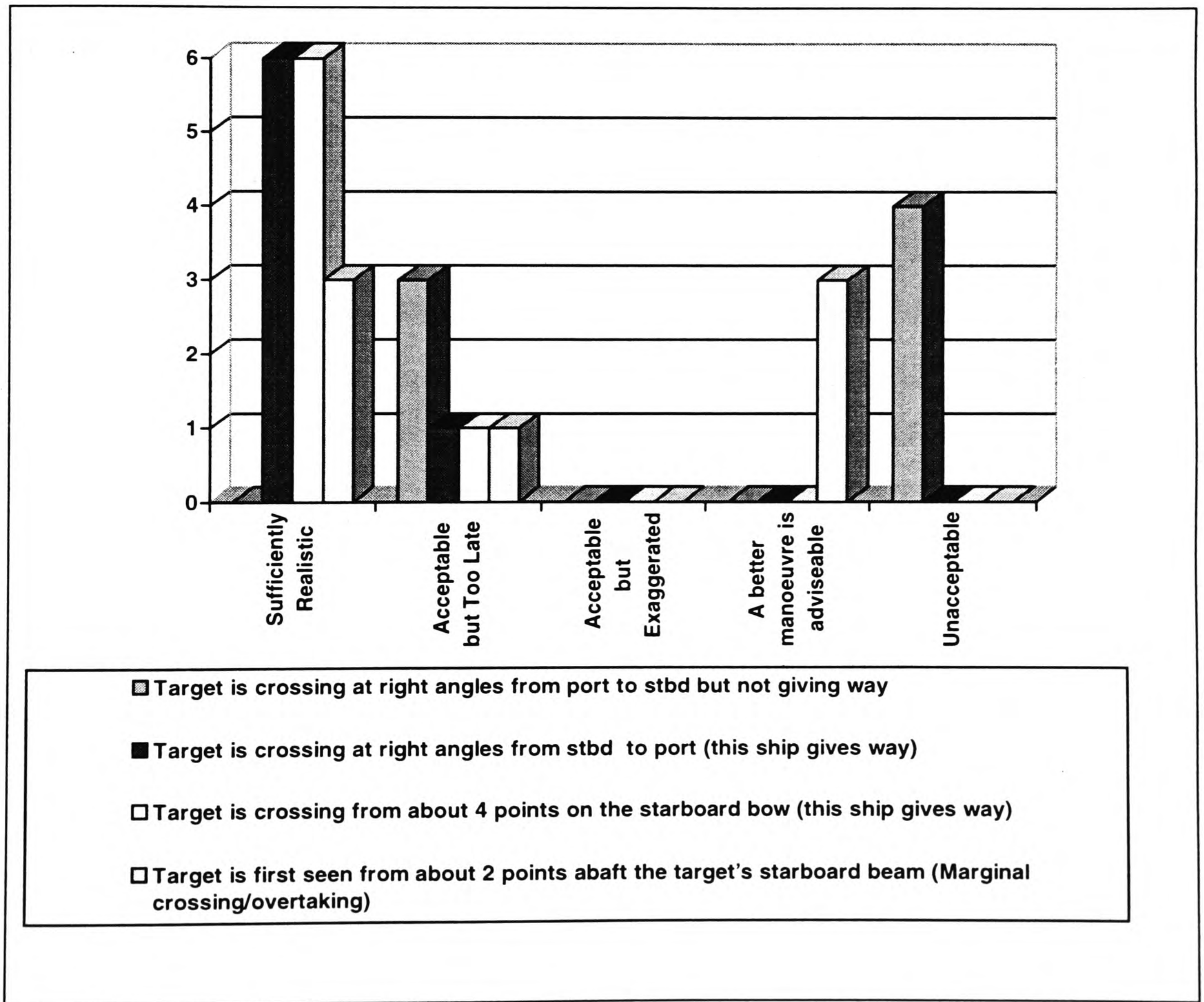


Figure 9-5 Crossing situations without current affecting the ships

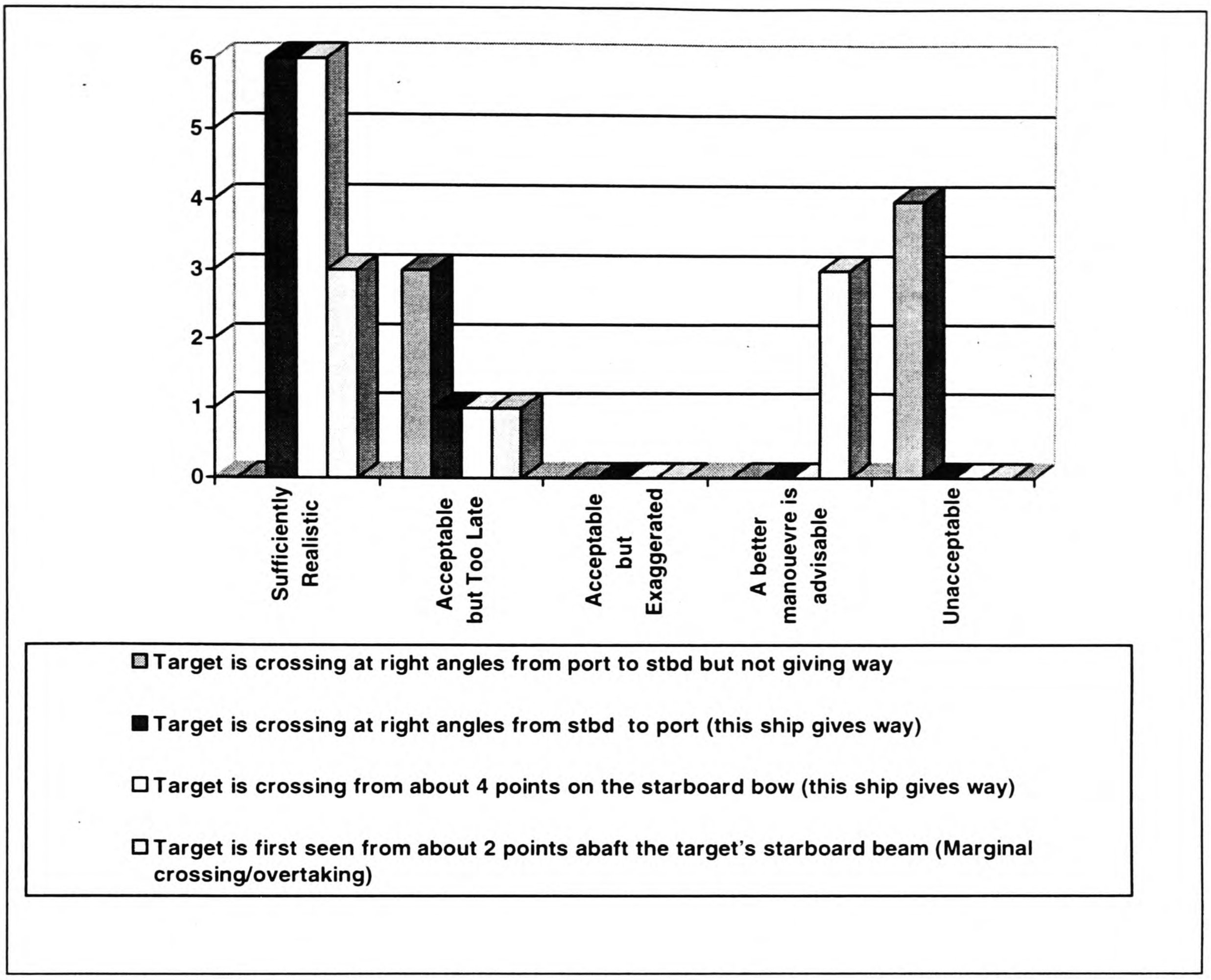


Figure 9-6 Crossing with current affecting both the ships

The overtaking manoeuvres were once again considered to be realistic for the own ship. These results were only obtained once the system had been tuned and the improvements discussed in 9.2.3.1 had been made to the software. In situations where the own ship is the being overtaken by a vessel that does not give way, the reactive manoeuvre is once again too late.

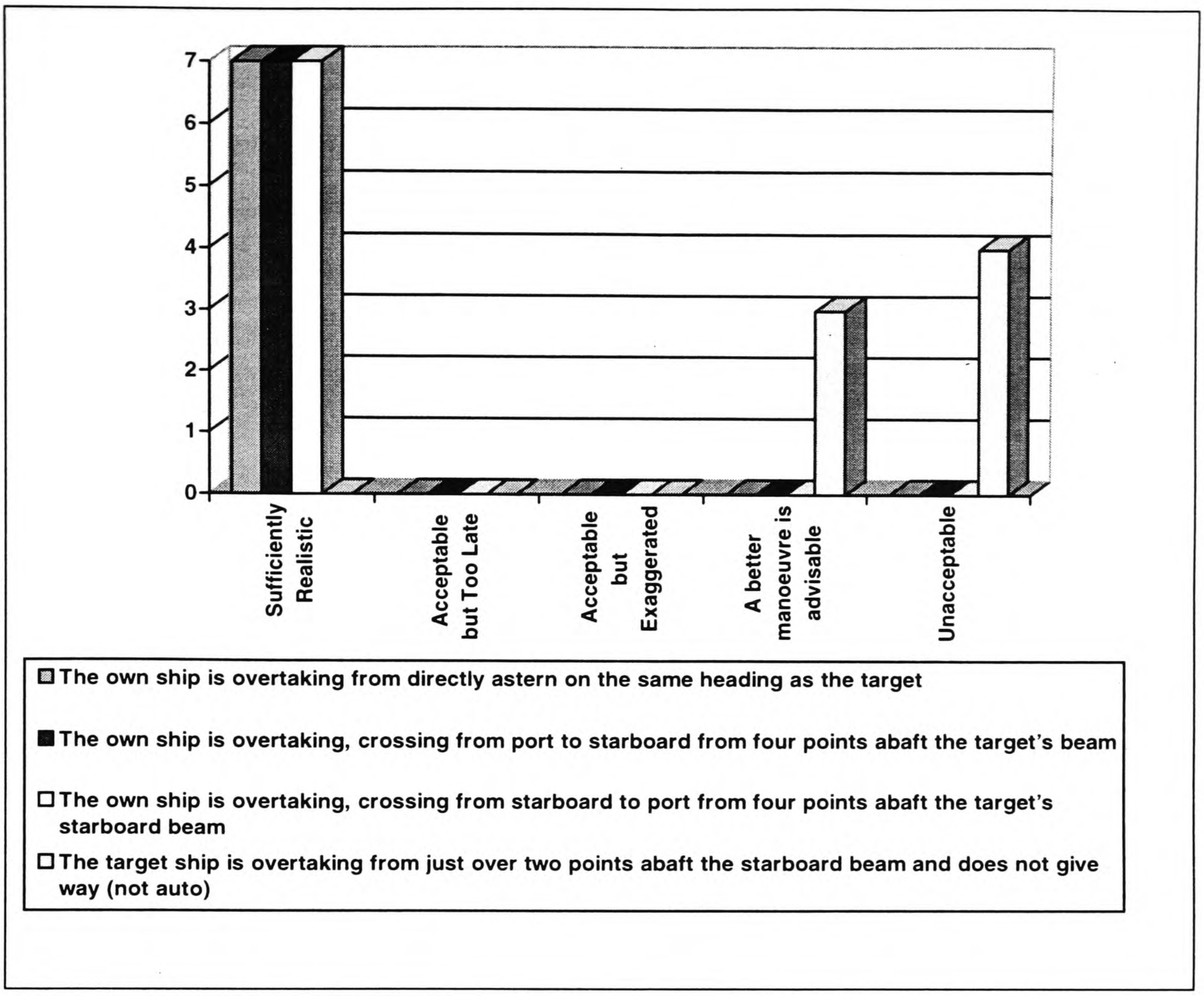


Figure 9-7 Overtaking without current affecting the ships

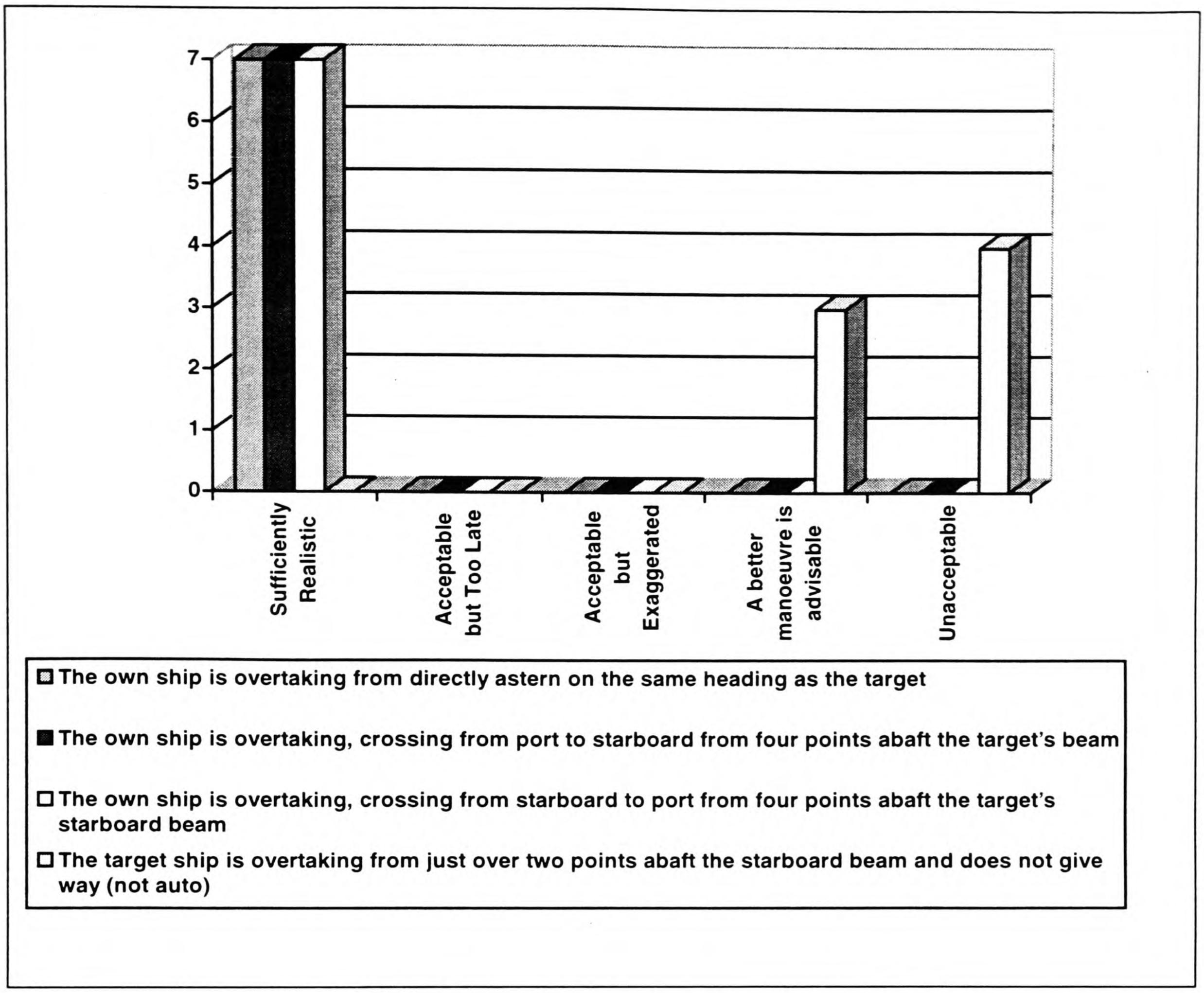


Figure 9-8 Overtaking with current affecting both the ships

- Overall Opinions of Track keeping and Collision Avoidance

Section 4 of the questionnaire, in Appendix C, contains questions about the value of the track keeping and collision avoidance to a simulation. All except one of the replies were positive that all of these features enhance the realism of the simulation. The other stated that these features were not desirable because it was too difficult to force the student into a specific course of action. In less formal discussions most of the other interviewees also expressed the opinion that the design and implementation of these features requires further research. The users would like to have the features available but also have an easy manual over-ride facility. It was, however, understood that MARINES is a test environment and that HCI[6] issues have not been fully addressed.

Furthermore, although agreeing that the implemented features could be useful, the users also expressed the opinion that multiple ship collision avoidance would be an improvement.

- The future of agents in Simulation

Unfortunately, while some discussion took place about the possible future of agents in simulation and most of the respondents were positive, few felt comfortable about answering section 5 of the questionnaire.

The ability to launch and stop agents dynamically (section 5 question 1) was considered important by one respondent; *"Yes - the more challenging the scenario the better"*. However, it was considered too complicated for the instructor to perform in the current system by another and the remainder did not answer the question.

Tugs (section 5 question 2) were considered an important issue by five of the interviewees. However, while they agreed in principle, they felt that they would like to see evidence that agents could perform these complicated actions before committing themselves.

Only one reply was forthcoming about fleet manoeuvres (section 5 question 4), a relatively long discussion about the capability of agents to perform the manoeuvres. Again it was felt that some evidence that this was possible would be needed.

Only two responses were obtained about car driving (section 5 question 4). It is seen as important by one respondent and probably too difficult by another.

It was generally considered that the questions in this section were looking too far into the future and it was difficult to determine whether agents were the long term solution to these problems. Upon reflection this is probably the case, certainly other questions considering shorter term possibilities might have elicited more response. It is also possible that the failure to provide optional responses for these questions left the

[6] Human Computer Interaction (Schneiderman 1987)

answer too open. In the other sections, although the respondents often provided their own phrase, the answer tended to focus upon the subject of the chosen option letter.

However, there were several less structured conversations on the future of agents in simulation that provided valuable possibilities for the future.

- The users' assessment of the manoeuvring for the set and drift of the current

It can be seen in Figure 9-3, Figure 9-6 and Figure 9-8 that the introduction of current has had little or no effect upon the assessment of the users, as to the correctness of the manoeuvres. Although the results are for a limited subset of the possible scenarios this is very positive. The agents have provided equally workable manoeuvres for two ships whether a current has been affecting the ships or not.

- Summary of the user evaluation

The different ship models were found to be at least adequate for target ships by some respondents and, more generally, realistic. The track keeping on its own was found to be acceptably realistic. Most of the users were already aware of the problems of set and drift, however, correct motion was considered desirable rather than a critical issue. After some demonstrations of the difference in apparent course, speed and aspect it was agreed that it was more important than it had at first, appeared. The difficulty of creating exercises that forced the student into a specific situation was raised. It became apparent that exercises incorporating variable current are seldom used by most instructors. Where current is used the exercise has normally been designed to take this into account. The target ships in such exercises following reasonably straight tracks when approaching the own ship, thereby minimising the effects.

For two vessels, the collision avoidance manoeuvres have been found to be adequately realistic for most overtaking and head on conditions. The manoeuvres for crossing scenarios were generally acceptable, but perhaps a little late. The reactive manoeuvres that the agents make when another ship fails to manoeuvre correctly under the regulations need to be reconsidered. The ability to respond dynamically to the changing situation was found to be both realistic and desirable.

The overall impression is that automatic track keeping and collision avoidance do together provide features that are valuable. There are desirable improvements such as multiple ship collision avoidance, but the basic features are attractive. To take full advantage of these features other parts of the simulators also need to be improved. The ARPAs on some micro simulators were criticised by one respondent, the accuracy of the ship models by another.

9.3 Combining the experimental results and the user responses

One or two users questioned the need for the collision avoidance and track keeping for target ships, preferring the traditional straight line tracks. However, when assessing the results of the exercises most responses were positive. The survey results are considered in combination with the experimental results in this section.

9.3.1 Strengths of the track keeping

When operating on its own the track keeping appears to be very robust. The set and drift of the current appears to have little effect upon the agents ability to maintain the track. The agent responding reasonably rapidly to the changing situation. The resulting track has been judged as adequately realistic for a target ship by all the respondents. This tied in well with the experiments where the track keeping approximately matched the desired motion.

9.3.1.1 Simulating different track keeping abilities

Unlike the target ships of a conventional ship simulator, such as the Maritime Dynamics Simulator at Kaohsiung Taiwan, the track keeping behaviour of each ship differs according to a large number of factors: the ship manoeuvring characteristics, the auto pilot settings, the agent's knowledge of the ship's manoeuvring characteristics, the set and drift of the current, the frequency of the agent's position fixing and each agent's perception of the environment. These factors mirror those found in the real world and produce complex interactions.

The use of these real world factors allows the simulation of typical ship motion. The agent applies corrections to the ship's desired heading in response to off track distance and course made good. The agent will respond gradually to changes in the set and drift of the current. Therefore, as shown in chapter 8, the 'aspect' of the ship is correct when viewed from another vessel.

A super tanker is unable to manoeuvre as a fishing vessel, taking longer to turn and stop. The manoeuvring of such diverse target models in MARINES can be seen in chapter 8. However, the majority of simulators do not make realistic allowances for this; the additional work load for the simulator instructor being high if several target ships are needed. In the simulator at Liverpool John Moores University target ships are affected by current and the models are reasonably realistic. However, exercises containing several target ships are not often performed, particularly where each ship is to follow a complex track. In discussion with the research group at John Moores it was also agreed that instructors needed to be highly competent and thoroughly familiar with the simulator in order to operate it, although the need for more than two or three target ships for training exercises was questioned.

One alternative is to employ more than one target controller or instructor, each controlling one or more targets. This approach has been used for the problem of air traffic control training. An example is the simulator at Ottawa International Airport, in Canada, that was visited as one of the Summer Computer Simulation Conference (SCSC 1995) activities. This simulator provides a highly realistic simulation and has the added advantage that the student air traffic controller communicates with several different respondents to radio messages. Unfortunately, the cost of training a single student is high, several instructors being required. It is doubtful that agents can replace these instructors in such a simulation, response to natural language being beyond present agent capabilities. However, agents have been suggested as an aid to the actual controller (Findler 1991).

9.3.2 Desirable improvements to the track keeping

The track keeping appears to work quite consistently. However, on a real ship, it is unlikely that a navigator will make all alterations of course using the auto pilot and this could be improved upon in the simulation. The other area that has been identified for improvement is the need for the agent to plan ahead so that if the end of the track is approaching, or a turn is required, that will need a reduction of speed the agent will respond accordingly.

9.3.2.1 *Auto Pilot*

In the real world PID auto pilots are normally only used for small alterations of course in deep water conditions. There are several reasons for this: large error signals may create unnecessarily large rudder motion; a navigator can adjust the helm directly to control the rate of turn more accurately, preventing large angles of heel and producing more accurate turning circles; genuine PID controllers can produce large off course integration errors, causing erratic behaviour; etc. Therefore, manual steering will be selected for more accurate track keeping.

The increased message traffic needed for a realistic simulation of an agent steering a ship manually would require a high communications bandwidth and a fast processor. On slower machines, such as a Pentium 75Mhz, DDE messages are already lost if the system contains more than ten ships and twenty agents, or is run faster than world time. Therefore, the agents in MARINES v1.07 do not have the ability to steer the ship using manual steering. However, creating a computer program able to alter the ships course in a more accurate manner than a PID controller has been considered by several researchers; fuzzy logic controllers have been suggested as one possible route to improved auto pilots (Polkinghorne et al. 1992) and track keeping (Burns 1995). An auto-pilot based on fuzzy logic has been shown to make large course changes more rapidly than a PID based system. This was achieved using smaller rudder angles and fewer changes of desired rudder setting. This could provide more realistic steering without the additional message passing needed for manual steering.

9.3.2.2 *Tight turns*

If the turning circle of the ship is larger than the area of safe water required to make the turn then the ship leaves the track. A warning is issued and the ship will return to the track after a short interval. In a production version it may be necessary to prevent the user from entering tracks that the ship is unable to follow at the desired speed, or increase the agent's intelligence to permit reductions in speed or even the ability to turn short round[7].

9.3.2.3 Short tracks

If the tracks that are entered do not permit the ship sufficient time to settle on one track before requiring another then the track keeping agent will prevent the subsequent turn. Once again a production version would require some mechanism to prevent this from occurring, this could take a similar form to that needed to prevent tight turns as mentioned in the previous section; preventing the user from entering illegal tracks.

9.3.3 The strengths of collision avoidance

The user evaluation shows that for collision situations involving two ships the collision avoidance in head on situations is seen as being highly realistic by all the interviewees. The overtaking situations are seen as being highly realistic by all the interviewees except one; the error that was discovered has now been corrected. (Unfortunately it has not been convenient to re-run the exercises with this interviewee in attendance.) Crossing situations where the own ship was the give way vessel under the rules were found to be highly realistic by all except one interviewee who considered the action to be too little, too late, although he agreed that many navigators did behave in this manner at sea. Therefore, the overall appraisal of collision avoidance between two ships is very favourable. This was considered adequate for the experiments in MARINES that generally involve only two ships.

9.3.4 Improvements needed for fully automatic collision avoidance

A more sophisticated determination of the danger is needed. Additional factors such as the relative speed of the ships and the size of the alteration that is required should be taken into account. An agent on a very slow ship will need to alter course earlier to avoid faster traffic in many situations. Also, if the ship is running at reduced speed, the steering is less responsive and the yaw due to the sea state and the swell is

[7] The term short round describes turning a vessel in a restricted distance. This is done using the engines and steering. On a ship with a conventional right hand turning propeller (left hand astern) the turn is usually made to starboard to take advantage of the transverse thrust.

generally greater. This reduces the accuracy of the information inferred from the historical data. As the accuracy of the course and speed of an approaching target ship is reduced, the agent will need to make a bolder alteration in order to ensure that the passing distance is adequate.

The effects on the ship steering are modelled in MARINES, according to speed. However, the agent does not adapt its beliefs for the ship behaviour. This is one of the reasons why it was detected that the action for a crossing situation appeared to be taken rather late; in the exercise the give way ship was running at a reduced speed of ten knots, while the stand on vessel was at full speed. The use of META-rules may be necessary to permit the agent to adapt for the prevailing conditions. The higher level rules could direct the agent as to when to apply the collision avoidance rules.

Multiple ship collision avoidance has not been undertaken. Some exercises in chapter eight have shown that a succession of one to one encounters have been successfully undertaken, but that several ships in close proximity can lead to agents making manoeuvres for one ship that place them in equal jeopardy with another. It was believed that this could lead to continual swapping of the most dangerous target, forcing the agent into a loop, iteratively, applying a manoeuvre to avoid one target and then the other. The problem is similar to that discussed by Penders (1993) for robots avoiding each other. However, this has not occurred in practice, a fact that may be attributable to a rule set that predominately selects a starboard alteration, for the power driven vessels considered in MARINES.

9.3.5 Realistic motion

It is the dynamic adaptation to a changing environment that is desired in MARINES. For this reason the user evaluation has also assessed the motion of the ships under changing conditions. Subtle changes to the set and drift of the current affect the state of the simulation. This can change the manoeuvring responsibilities of the ships' navigators to each other. For example, a head-on situation may become a crossing situation, as described in Chapter 8. This change may have knock on effects on other meeting situations and, to be realistic, each ship should be manoeuvred accordingly.

On conventional simulators this has only occurred if the simulator instructor has been very experienced and prepared to perform manoeuvres for all the affected target ships. In MARINES the instructor may have to manoeuvre the ships in scenarios where a number of ships all need to manoeuvre at the same time, to avoid each other. For multiple situations, each involving two ships, the MARINES simulator has been demonstrated to perform most of these manoeuvres automatically.

9.3.5.1 Collision avoidance, track-keeping and set and drift combined

The track-keeping agent advises the collision avoidance agent of the course to steer to maintain the desired track. Small corrections of course are applied to maintain the track, counteracting the set and drift of a current and returning to course after a collision avoidance manoeuvre.

Using the MARINES test bed, the results of scenarios run when a target ship is affected by the current have been compared with those when a target ships was not affected by the current. This has displayed the advantages of the system when exercises include the set and drift of the current : the aspect of the approaching vessel is displayed very much as it would be seen for a real ship at sea; the agent adjusts its decisions and actions dynamically, according to the presiding state of the simulation; as in the real world, the ships' tracks are affected by changes to current, responding in a reasonably natural manner, by applying adjustments to the course steered; the CPA, TCPA course and speed of the approaching vessel are calculated very much as they would be on a real ARPA at sea.

9.4 Using the MARINES technology in a commercial instructor station

On most simulators, simple exercises involving one or two target ships on steady courses are relatively realistic and easy for an instructor to control. These straight forward exercises form the important basic training needed by navigators. The MARINES technology does not greatly enhance this style of application. However, if a simulator is to provide exercises that include the set and drift of the current and many target ships then there is no doubt that the effects should be realistic. At best, unrealistic aspects of exercises will be ignored by the students, at worst, they will serve to misinform the students and the STCW(1995) code requires simulators to be sufficiently realistic. The motion of the ships in MARINES has been judged to be a reasonably accurate facsimile of the real world by a number of specialist users. The application of the MARINES track keeping and limited collision avoidance to future simulators can provide solutions to multiple situations involving only two ships. This should permit slightly more complex scenarios, involving the set and drift of a current to be performed successfully. Additionally, changing the current at run-time will alter the behaviour of the target ships in a realistic manner.

Two important challenges must be addressed before the maximum benefit can be gained from the use of the MARINES technology in an instructor station on a real simulator. The first is more complete collision avoidance; there are some improvements needed for two ship avoidance, as already discussed, and then multiple ship interaction needs to be addressed. The second challenge is to improve inter-agent negotiation, providing more feed back and addressing dangerous alterations of course into areas of shallow water.

In order to create a commercial system more quickly it may be necessary to reduce some of the complexity. For example, the ships in MARINES are intentionally affected by yaw due to the state of the sea, historical data is affected and thus the manoeuvres are sometimes based upon inaccurate information. This, as in the real world, can lead to erratic behaviour, perhaps not what the instructor requires.

The user interface of MARINES has progressed alongside the changes to the system functionality. Some of the choices have suited the MARINES experimenter rather than the future instructor of a real system. Once the designs of the agents have reached full maturity, research is needed as to how best to harness the additional functionality provided by the track keeping and collision avoidance agents. Novice instructors need to be assisted and experienced instructors need to be given the freedom to use the system without continual interruptions from the agents.

Judging from the results of the survey, once the technological challenges have been addressed more fully it will be necessary to work alongside users in their operational environment; possibly with a conventional instructor station and a MARINES style instructor station available. The comparison of the two would help to identify more accurately where the agents could provide help without hindering or annoying the instructor, and where the agents are really helpful. The progression of students should also be monitored from a teaching and learning viewpoint, if possible comparing control groups taught by the same instructor, controlling the same simulation exercises, and using different instructor station technologies.

There is some way to go before a completely automatic bridge simulator, with realistic target ship motion in a dynamically changing environment is available. So far, the MARINES project has taken an incremental step towards that goal, using a Multi-Agent Systems architecture.

9.5 Summary

This chapter has discussed the evaluation that has been performed with a limited number of experienced simulator users and designers. The evaluation was carried out over several months and some system development was performed in parallel. These developments were generally in answer to suggestions from the interviewees. It is believed that this prototyping style of development has been beneficial in this research; many errors and omissions having been eliminated early in the development process. The opinions of the expert users have generally supported the manoeuvres as being reasonably realistic for the scenarios tested. Although, the timeliness of the manoeuvre for certain crossing and stand-on scenarios has been questioned.

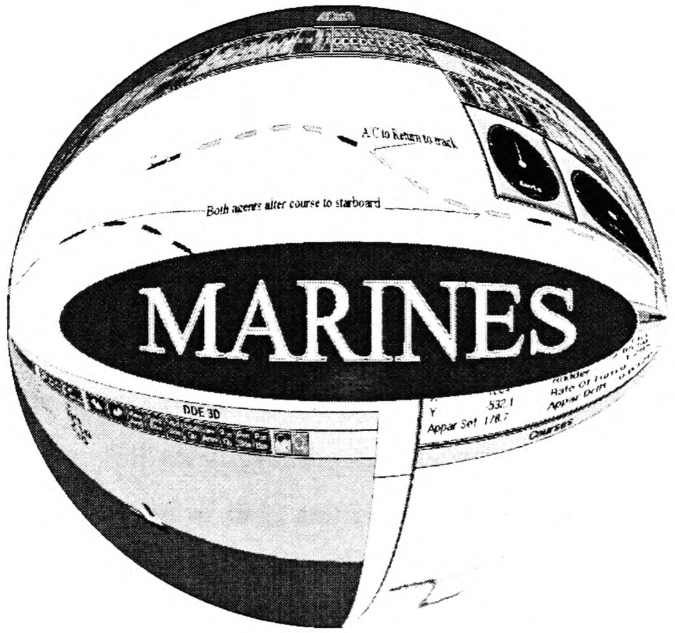
The results of this evaluation and the results of the exercises that were performed in chapter 8 have been compared. The ship models and track keeping have been found to be sufficiently realistic for the target ships and, after tuning, the collision avoidance mechanism has been found to be generally adequate for situations involving two ships. The motion of the ships has been adequately realistic. The CPA, TCPA, course and speed of the target ships have been demonstrably correct in many experiments.

In order to use the techniques in a commercial project some further research is desirable. Automatic collision avoidance expert systems for multiple ship collision situations have been created in other research projects, and have been shown to be

plausible. The STCW '95 code implies that this realistic motion will be necessary if simulators are to meet the requirements in the next century and be used for more than simple exercises.

The MARINES technology will not replace the need for an instructor on a marine bridge simulator, however, it should go some way to assisting the instructor and providing additional functionality and realism.

Chapter Ten



Concluding Remarks

10. Concluding Remarks

10.1 Overview

The main points raised by the research are re-stated at the beginning of this chapter. Then, in section 10.4 the motion of the target ships that has been achieved is summarised. In section 10.5, agent assistance for the instructor is considered. The robustness achieved and the flexibility of the system are discussed in section 10.6 and issues such as recovery from errors and changing the agent characteristics are highlighted. Section 10.7 looks at the issues surrounding the use of a Multi-Agent System in a simulator, highlighting some crucial areas that need to be addressed in order to create a successful simulation. The conclusions drawn are listed in section 10.9. Finally, the chapter and the findings are summarised.

10.2 Introduction

The environment that the MARINES agents exist in attempts to model the problems as they occur in the real world, although, of course, this is done in a simplified manner. The agents receive information based upon that which a real officer of the watch would have. For example, the agents store the bearing and range of any approaching target over a period of time and extrapolate the results to determine a plan. Therefore, the reasoning is based on imperfect information, the agent may be unaware of a manoeuvre by another ship. The accuracy of the inferred results is affected by natural constraints, the ships yaw slightly in the sea way, preventing a completely accurate assessment of the situation. If one of the ships alters course the calculated information becomes quite inaccurate in a similar way to a vector on a real ARPA. Therefore, when running the same exercise several times there will be subtle differences, as in the real world. If a current is introduced, affecting the ships, then the agents will adjust accordingly, applying corrections to maintain track. The adjustments may produce different collision avoidance responsibilities and the agents will alter their plans and manoeuvre accordingly.

The MARINES research has demonstrated that a MAS architecture can be used to build a simulator with reasonably realistic target ship motion, including track keeping and some collision avoidance manoeuvres. The complex behaviour of the simulation derives, not from complex heuristics approximating to the desired results, but rather from modelling the individual characteristics of simpler processes and letting them interact (Minski 1986). For a number of relatively simple experiments, this has been shown to create a simulation that responds naturally to environmental changes.

10.3 Brief Recapitulation

The manoeuvring characteristics of the computer generated target ships have been tested and shown to approximate the generic ship type that they simulate. The models are not of the complexity or accuracy presented by Pourzanjani(1990a) or McCallum(1980) for the own ship, but do provide a more accurate model than the computer generated target ships used on micro simulators such as the Mardyn Ship Handling Simulator in Taiwan.

Automatic collision avoidance has been implemented in MARINES and shown to work reasonably realistically for a wide variety of situations. Some situations, particularly those where the stand on vessel has to make an emergency manoeuvre require further consideration. In general, the ship models previously used for collision avoidance simulation have not been subject to natural constraints and inaccuracies of tracking a ship at sea. MARINES does model some of these features, the yaw of the ships in a sea way, overshooting the desired heading when altering course and the manoeuvring characteristics of the different vessels. The characteristics of the collision avoidance agent are also modelled, the agent may take early or late action and use different rule sets. In chapter 8 a number of collision avoidance scenarios have been tested, reflecting a selection of head on, crossing and overtaking situations. The results are not as black and white as some previous computer simulations of collision avoidance, this reflects the uncertain nature of real situations.

The track keeping algorithm has also been shown to give quite realistic effects. The curves flow and the slight deviation of ships from their desired tracks is similar to the real world. The use of simplified PID auto pilots and ship models with three degrees of freedom permits subtle changes to be modelled. For example, the helm and counter helm controls can be adjusted to change the steering characteristics.

Considerable research has already been done elsewhere into creating automatic collision avoidance for ships. In particular, there are a number of similarities between MARINES and research by Blackwell et al. (1989). Similarly to MARINES, Blackwell's system considered collision avoidance between two vessels. Later, Blackwell and Stockel (1990) consider multiple ship collision avoidance for groups of ships, and Blackwell et al. (1991) consider multiple two ship encounters. MARINES can cope with some multiple two ship encounters, to a slightly lesser extent than Blackwell et al. However, MARINES introduces track-keeping and current to the equation, and the individual target ships each have their own intelligent agent, permitting characteristic behaviour to be modelled.

Research interest has also been shown in automatic track keeping. However, while theoretical studies have discussed integrated navigation systems, experimental test beds have normally specialised in one or other discipline, not the combination of the two. The closest track-keeping style to MARINES is that used in-house at Maritime Dynamics; this uses mathematical routines but goes further in modelling complex shallow water and bank effects.

Although the PC Maritime 'Officer of the Watch' simulator has combined automatic collision avoidance using domains with direct linear tracking motion of computer target ships, no environmental conditions affect the ships; the ship models are simple and they follow the desired tracks precisely apart from when avoidance is needed.

Mathematically modelled computer generated target ships, with three degrees of freedom, automatic track keeping, automatic collision avoidance and Multi-Agent Systems are all combined in MARINES.

10.4 Motion of the computer generated target ships

Guicharrousse(1990) has criticised computer simulations that *"will indefinitely produce the same results in answer to the same manoeuvres. It is very different from the real world where a manoeuvre will never recur in the same way"*. The STCW(1995) also requires realistic behaviour. Although Guicharrousse's requirements go much further than those tackled, one of the objectives of the project was to determine whether the motion of the computer generated target ships could be realistically affected by changes in the environmental conditions, manoeuvres of the other ships and characteristics of the agent navigators.

10.4.1 Changes to the set and drift of a current

If the set and/or drift of a current are altered at run time, this should dynamically affect all the vessels in the simulation and, therefore, their collision avoidance manoeuvres. This has been demonstrated in MARINES in a number of ways.

10.4.1.1 Speed and course calculations

As shown in chapter 4, the ARPA calculations and aspect of approaching ships on many simulators are erroneous when current effects are present. The exercises in chapter 8 have demonstrated the value of the correct results and support the need for more realistic target ship motion.

10.4.1.2 Ships heading

In chapter 8 and chapter 9 the agents have been shown to alter the desired course in order to track-keep in response to changes in the set and drift of the current. The 3D view displays a gradual change of aspect as an approaching target ship under agent control responds to the changing set and drift.

10.4.1.3 Deviation from track

In the same way as navigators in the real world, it takes a little time for the agents to respond to changes in the set or drift of the current. Therefore, the ships deviate slightly from the desired track. The agents apply corrections to the desired course to achieve the change. The rapidity with which the agent responds and applies these corrections affects the future position and track of the ship.

10.4.1.4 Collision situation

Additionally, the change in the ships' headings has been shown to alter the collision situation. For example, head on situations become crossing situations. When this occurs the agents respond at run time to the changes, applying different rules according to the changed responsibilities.

10.4.2 Alteration of course under agent control

Guidance on when to alter course to follow the desired track is given to the collision avoidance agent by the track-keeping agent. As in the real world, the accuracy of this depends upon: the manoeuvring characteristics of the ship; the adjustment of the auto-pilot controls; the agent navigator's perception of the ship's manoeuvring characteristics; the tightness of the turn; and the presence of other traffic in the area. The resulting track of the ship does not follow nice mathematical curves, nor is the rate of turn of the ship constant. The ship may overshoot or undershoot the desired track and return gradually onto the heading and track.

10.4.3 Motion of different ships

Each different ship model in the simulation manoeuvres in a characteristically different manner. This has always been the case for the models controlled by the students. A few simulators also have realistic ship models for the target ships, but, where this is the case, automatic track-keeping and collision avoidance have not been provided.

10.4.4 Overall analysis of the ships' motion under agent control

Changes in the environmental conditions have been shown to alter the behaviour of the ships in a manner that has subtle and realistic results; the effects of changes to the set and drift of the current, different ship models and different agents all contribute to a sophisticated model of computer generated ship motion.

There are some omissions within the guidance capabilities that need to be addressed. These fall into two main categories:

Firstly, the collision avoidance mechanism does not cope with all situations. Multiple ship situations have not been catered for and a small percentage of two ship situations have not produced acceptable results. It is felt that this is beyond the scope of this project; several other research groups are gradually providing a solution in this area.

Secondly, the quality of the land avoidance needs to be improved. The track keeping algorithm does provide a warning if the ship deviates from the route by an unacceptable margin. This enlists the instructor's assistance for dangerous situations. However, realistic land and shallow water avoidance needs to be addressed in more depth, both for simulation and for the production of fully integrated intelligent ships' bridges.

There is scope for considerable further research. Nevertheless, limited consultation with experts in the field of ship dynamics has shown the resulting ship motion to be similar to the motion that would be expected from a real ship. Therefore, ships can be modelled, using a MAS, to be realistically affected by changes in the environmental conditions and characteristics of the agent navigators.

10.5 Instructor Assistance

The agents provide the instructor with assistance in two areas. Interactive exercise development and monitoring procedures.

10.5.1 Interactive exercise development

As shown above the agents co-operate to maintain the track and avoid other ships. This permits the instructor to change the courses and speeds of the ships interactively, without unduly worrying that the computer generated ships will hit each other. For example, consider the case where a student decides to slow the own ship down. The student ship will arrive at a position after the target traffic has departed. The instructor may need to respond by reducing the target ships' speeds, altering the target ships' future courses or even introducing a new target ship, so that the student is still presented with the desired problem.

10.5.2 The agent monitor

The agents monitor the situation and alert the instructor to infringements of domain and increasing danger. If the instructor wishes the agent will then centre the simulation over the danger area. The instructor does not need to monitor the entire simulation, rather, it is possible to concentrate on a single area, for example, the student ship's progress.

10.5.3 Overall analysis of instructor assistance

The ability of the agents to alert the instructor of danger between ships is often valuable. However, there are some shortcomings and spurious messages sometimes occur. For the MARINES technology to be really useful multiple ship collision avoidance need to be addressed and the HCI aspects need to be considered. Some simplification of the system might also be necessary to make the agent responses more predictable. The agents in MARINES solve the problem as a real navigator might, however, additional information is available, that could simplify the task. For example the course and speed are calculated, by the agent, from historical information, but they are available directly from the environment. This could possibly decrease the absolute realism slightly, but would make it easier to model the problem. There is still some way to go before MAS simulation will provide an instructor with truly intelligent assistance, but the potential has been demonstrated.

10.6 Robustness and Flexibility

10.6.1 Floating Point Errors

To demonstrate the resistance to errors, floating point errors have been intentionally introduced terminating a collision avoidance agent process without stopping the simulation. This has been shown to work in most cases, although some errors will terminate the simulation.

10.6.2 Re-configuration

Shortly after the collision avoidance agent is terminated the track-keeping agent, that has been advising it, will reconfigure itself to send the ship messages directly. Thus, some functionality will be lost but, in most cases the simulation will continue to run.

10.6.3 Agent Characteristics

Collision avoidance agents have been created with minor differences in their 'rule of the road' knowledge bases. This has shown the flexibility of the agent approach. An agent that models a common misconception can be attached to a ship. An instructor may have difficulty remembering the characteristics of all the different ships, thus an agent can possibly be more consistent.

10.6.4 Dynamic introduction of new agents

Agents can be introduced to the system dynamically at run-time. This permits new agents to take over the navigation of a ship, as and when they are needed. Additionally, new and different characteristics can be introduced.

10.6.5 Overall analysis of robustness and flexibility

It is virtually impossible to eliminate all the errors from a large software system. One of the major fears of the instructors in Taiwan discovered by Chen(1992) was the fragile nature of the simulators; the instructors were not confident about running the simulator because it often failed. If a run-time error occurred in the instructor station software the entire simulation would terminate. Research in MARINES has shown that errors in non-critical functions can be tolerated and the software re-configured to continue running with slight loss of functionality.

The flexible nature of MARINES has shown that agents with different characteristics can be launched dynamically. Subtle changes can be introduced into the simulation through the use of agents with different rule sets for different vessels.

10.6.6 The complexity

As well as helping the simulator instructor the use of agents has made it possible to add features. MARINES is a research project and this added functionality has been used to improve the motion of the computer generated target ships. This added functionality has necessarily made the operation of the instructor station differ from a conventional system. When creating a system for a full simulation the balance of complexity and additional functionality will need to be addressed. As stated in one evaluation interview *"a default configuration should be available for novice instructors"*.

10.7 Discussion and Experiences of the use of Multi-Agent Systems in Simulation

Based upon experience in this research and other simulation projects, several factors should be borne in mind when considering such a MAS for commercial simulation purposes, some of these are discussed in this section.

10.7.1 The Multi-Agent System Paradigm

Agent Orientation and Object Orientation are closely related. Agents are to an extent Intelligent, Communicating, Co-operative Objects. Features of Object Orientation such as encapsulation, inheritance and polymorphism are also inherently part of an agent. This permits multiple instances of agents, with similar functionality and unique features, to be easily created; each agent is particularly highly cohesive and the instances are coupled only by the high level messages.

The MAS paradigm has proved generally useful in the abstraction of the problem. The considerable research that has been performed in the field of Multi-Agent Systems has been beneficial in designing a simulator instructor station that includes automatic marine collision avoidance and track-keeping, in a way that has maintained the discrete nature of the components. Thinking of the objects in terms of intelligent, co-operative processes with relatively high level communications led to consideration of how the human navigators communicate and co-operate.

Pragmatically, there is a need to make use of the parts of this MAS paradigm that are important to your system, and eliminate the parts that, although useful elsewhere, will not assist in a particular problem area. If agent systems are to be useful commercially the system should model the problem, not be driven by the need for 'Agency'.

The MAS paradigm is a tool that solves a class of problems; generally where the complete problem area consists of a number of easily identifiable sub-areas and the interactions between the sub-areas create a complex behaviour, that is hard to understand. This being so, the MAS model should reflect reality; use intelligent components to model the parts of the system where a discrete solution is hard to provide and use mathematical or functional models where they are more apt. 'Weak' agency may not be the long term goal of the advocates of MAS, but it may solve some immediate problems. Success in this area may well support further research, rather than undermining the ideals.

Of course, there is still a need for systems to research into MAS communication techniques and frameworks, where a high communication level may be desirable. Therefore, some research systems will still need to include loquacious agents (Staniford and Paton 1994).

10.7.2 The Agent Implementation

At the implementation level the value of agents is less clear. On a single machine Object Orientation, dynamic binding and dynamic link libraries may, together, offer most of the flexibility found in MARINES, without the need for inter-process communications. Creating a MAS will normally require more programming effort than a single program.

The implementation of an Agent Oriented Programming language and an interpreter (Shoham 1991) for the communications appeared desirable, but, on the basis of rapid performance, information was passed using relatively low level message blocks. A higher level language could become essential if negotiation (Zlotkin and Rosenschein 1989) and joint-intentions (Jennings 1993), etc., are to be widely used. This would probably only be possible on a distributed system, or a far more powerful platform.

The current implementation does appear to have potential in two areas, the first, already mentioned, is resistance to some software errors. The second is that the system lends itself to distribution. The agents in MARINES all run on a single machine but this is implementation specific; they could well run on a network of separate machines.

Furthermore, the agents have been built up individually, this has reduced the danger of side effects in the other programs. In particular, this has been valuable from a research aspect. Many changes have been made and alternative algorithms implemented, as in any software maintenance this has occasionally produced unexpected results. In general, these problems have only been apparent in the implementation of a single agent, thus, they have been easier to track down and eliminate; each individual program being smaller and easier to understand.

10.7.3 Agent Debugging

The problem of debugging separate agent processes on a single machine can be quite different from debugging a monolithic program.

There are two common types of debugging, hard debugging where the debugger takes over the entire machine and prevents other processes from running, and soft debugging that tries to handle messages for all the processes, permitting the system to run normally. There are also two derivatives of Windows 3.1 the version sold commercially and a version that assists debugging. MARINES was debugged under the normal commercial system.

After completing the testing and debugging of an agent on its own some soft debugging is normally required. A single instance of the agent can be dynamically linked to the environment and tested, the environment supplies the information to the agent, just as it would in a normal run-time environment.

There are advantages and disadvantages to this. One advantage is that the environment simulation can be paused at places where the agent does not operate as expected. The agent can be stopped, corrected, re-compiled and re-run without the need for lengthy setting up procedures for the simulation. In a continuous simulation this also makes it easier to re-create the error that last occurred. The compile and link times for individual agents are also shorter than for the whole program. A further advantage is that the size of each agent program is small, so that the debugger will usually run without memory problems. A disadvantage is that the debuggers tested can only support a single instance of a process to be debugged. Therefore, when a large number of instances of the same agent are needed the debugger cannot be used; a monolithic program could contain many instances of an object and could normally be run under a debugger.

10.7.4 The message passing system

A central part of this research has been the production of a suitable message passing system. In MARINES this has been the most contentious issue facing the production of a robust Multi-Agent System with rapid response. This has been described in chapter 5.

10.7.4.1 Robustness

The underlying transmission layer of the inter-process message system has to be very robust. It should not be possible to crash the message system or corrupt other programs by passing illegal messages. As shown in MARINES it should be possible for one agent process to suffer a run-time error without introducing errors into other processes.

10.7.4.2 Bandwidth

In a real-time simulation the message passing must be rapid and the level of communications must be kept as low as possible. Careful consideration of the message content is needed to overcome this. For example, in the MARINES simulation the simulated GPS position fixing and a simplified track keeping map obviate the need for a high communication bandwidth for track keeping. The problem is exacerbated in any system that resolves around a single environment, the communications bottleneck for many systems. Therefore, the architecture of the simulation has to be carefully planned; wherever possible the agents and processes should communicate directly rather than via a central hub. Steeb et al.(1981) made an essential contribution, with their discussion on possible architectures. The MARINES architecture is a hybrid of the hierarchical and object-centred autonomous architectures and this appears to perform reasonably well. It is quite similar to a real world abstraction of the marine environment being simulated, however, it should be remembered that the communication on a ship's bridge is quite rigidly structured and this may not provide the best solution to all simulation problems.

10.7.4.3 Message Recovery System

On any truly distributed system with frequent communications there will be a high probability of message collisions. Many of these may be recovered by network software, however, the agents need a method of deciding how to prevent the system being flooded by a temporary increase in traffic.

Some messages may be urgent, some may have to be processed in order and a few may be redundant. In any case, in a real time simulation it is unlikely that responses that are delayed by more than a few seconds will be useful. As described in chapter 5 the messages in MARINES have been divided up according to urgency and failed messages are cancelled and then requested again after a random period. This was found to be essential as the DDE message stack would sometimes overflow if too many messages were passed simultaneously.

10.8 Future Research Possibilities

10.8.1 Negotiation for the Track Keeping and Collision Avoidance Agents

The navigators aboard ships have to solve problems that avoid fixed danger at the same time as manoeuvring to avoid ships. This will sometimes involve some form of negotiation between the navigators. A simplified example could be that the navigator in command decides upon a manoeuvre to avoid another ship and then requests a check to ensure that this will not take the ship too close to a fixed danger, such as shallow water. The navigator plotting the course must convert the desired heading to a land stabilised track and determine how long the ship could maintain that track before it would result in a dangerous situation. If it is felt that this is too dangerous the course plotting navigator might suggest a change in speed or a different heading. The final decision lies with the navigator in command, but the two negotiate with the joint goal of safe passage without undue delay. However, each has only partial knowledge of the problem space. Negotiation with partial information has been studied by Zlotkin and Rosenschein (1989, 1991), interestingly they also considered what happens if the agents lie to each other. While the navigators should not lie to each other they will probably err on the side of caution and also try to maintain a domain of their own choosing around the vessel. Joint Intentions (Jennings 1993) are formed when the two agents agree upon a single course of action. Both agents have joint responsibility to ensure that the action is successful. In this case, if the navigators are agents then, should the manoeuvre take longer than expected and cause a greater, and more dangerous, deviation, the agent performing the track keeping may well begin to loose confidence and re-consider its commitment to the plan. The responsibility model then determines how the agent will behave; possibly suggesting a new course of action.

10.8.2 Assessing Different Agent Architectures

The implementation of the agents in MARINES has been limited due to time and machine performance. Now that the test bed exists it would be interesting to create agents based upon different architectures and examine the relative merits. Many of the MAS test beds have been used for this in the past, however, most are either grid based or have discrete planning and action steps. Therefore, their world does not change asynchronously, in an analogue manner, during the planning stage. In MARINES if an agent takes too long over planning this may lead to a solution that is too far out of date to be implemented. Additionally, the distance a ship travels and the success rate of bold manoeuvres could be compared with agents that often reconsider their actions, etc.

10.8.3 Unusual Two Ship Collision Encounters

The interpretation of the qualitative nature of the rules has led to studies of ideas like safe passing distance. A major step forward came when the concept of domains (Goodwin 1975) was statistically proven. The theories of Arenas (Davis et al. 1980) and Regimes (Burns 1995) also provided useful concepts. Together, these concepts help to solve many of the problems encountered. Even so, the solutions do not model navigators' thought processes, rather they express the results of these processes for a large percentage of the problems. There remain several conditions where, even two ship collision encounters can be difficult to resolve in compliance with the regulations.

Consider the situation where a very slow moving vessel encounters a very fast ship; by the time sufficient information has been inferred, it may be impossible to maintain the desired domain by making a manoeuvre that complies with the rules.

A common example of this case is when a yacht under power, a power driven vessel under the rules, encounters a passenger ferry. The yacht may have a maximum speed of around 5 knots, and the ferry may be capable of in excess of thirty knots. For yachts without radar in the English channel a detection range of five miles or less is often the case. Therefore, the Ferry will reach its CPA within ten minutes. Assuming a three minute period to determine the danger the yacht can only progress about 0.6 nautical miles before the ferry reaches it. If the yacht is a crossing vessel with the ferry just abaft the beam on its starboard side and it has been determined that the ferry will pass too close astern, what is the desirable manoeuvre, and how can an agent achieve this? More to the point, as the stand on vessel, the ferry should maintain course and speed under the rules, however, the ferry is likely to wish for a larger domain than the highly manoeuvrable yacht. At what point is it *"so close that collision cannot be avoided by the action of the give way vessel alone"*[1] permitting the pilot of the ferry to take avoiding action under the rules?

From a simulation viewpoint what actually occurs in the real world is of equal interest. The pilot of the ferry will usually have a healthy regard for the danger presented by yachts and may well manoeuvre early to avoid a close quarters situation. This could possibly be construed as *"being required by the ordinary practice of seamen, or by the special circumstances of the case"*[2]. However, platitudes, such as this, are too ambiguous to code as production rules.

Further research is needed to determine a set of rules or another method of providing a realistic result. James (1986) suggests that *"such computer systems should incorporate simple models of the navigators' decision processes"* and that *"such behaviour should arise endogeneously through interactions within the model, rather than be imposed empirically"*. This fits in well with the Beliefs, Desires and Intentions(BDI) model used in some Multi-Agent Systems (O'Hare and Jennings 1996). Some beliefs are implicit in the collision regulations and the situation analysis; these form an integral part of the agents in MARINES. The agent already has to believe that the manoeuvre will be more successful than the present track, the next step is to investigate how to model the uncertainty of some situations, an area that has been the subject of MAS investigations(Sugawara 1993).

---

[1] An Extract from Rule 17 (IRPCS 1989)

[2] An Extract from Rule 2 (IRPCS 1989)

In order to achieve a truly realistic simulation, the different agents will also need some individual characteristics. The skippers of fishing vessels will have different beliefs, desires and intentions from the masters of super tankers or the commanders of nuclear submarines. This again supports the use of Object/Agent modelling techniques, where intelligent, polymorphic agents may provide the needed variety.

10.8.4 Extending Rule Sets and Manoeuvring Responses

Two rule sets have been tested in MARINES v1.07 the normal power driven vessel rule set and a rule set where the ship makes an erroneous alteration to port for some almost head on encounters, simulating a common manoeuvre seen at sea.

Research into different rule sets for different types of vessels and different mistakes made at sea would be valuable in creating realistic encounters on a simulator. This could have applications in a number of simulation specialisations; bridge training could be more realistic; fast time port design simulation could incorporate target vessels that simulated normal traffic flow into and out of the port, for example, if a chain ferry[3] makes a crossing, delaying all the traffic at a harbour entrance.

Vessels often behave in characteristic fashions; a yacht under sail may try to avoid an alteration of course that forces the helmsman to jibe; a fishing vessel may follow a particular depth contour; small, manoeuvrable, motor yachts may adjust speed, rather than course, to avoid other traffic; large merchant ships, in unrestricted water, usually avoid traffic through alterations of course in preference to changes of speed. The complex processes needed to model these characteristics accurately can probably only be provided by humans. However, individual software agents should be able to provide reasonable, and consistently characteristic, results for each ship in the simulation.

[3] Ferries crossing areas, such as Poole harbour entrance, are sometimes moved by the motor tugging at a chain attached to the shore at either end of their track. They are almost unaffected by the set and drift of the current, and their relatively sudden crossings may cause the traffic in the vicinity to make unplanned manoeuvres.

10.8.5 Multiple Ship Collision Encounters

The collision avoidance regulations (IRPCS 89) only discuss situations involving two ships. When more than two ships are encountered simultaneously, the navigator should attempt to produce a solution for all the ships that complies with the regulations. It is not always possible to do this and to maintain the desired domain. In such circumstances the navigator makes a compromise, producing a solution that should complement the manoeuvres the other ships are expected to make.

Blackwell and Stockel (1990) suggest a method of avoiding a group of approaching ships, all from the same direction. This covers a large number of common encounters where all the ships are proceeding in a traffic separation scheme or on a commonly used route. However, this is only part of the problem, in many busy areas the ships may well approach from different directions in such a way that a manoeuvre to avoid one ship will result in a close quarters situation with another. Smeaton and Coenen (1990) have produced a system that considers a complex situation involving five ships. Smeaton and Coenen (1990) and Smeaton et al. (1992) also propose the amalgamation of collision avoidance and navigation using ECDIS for an integrated ship's bridge. Smeaton et al. (1994) discuss the amalgamation of ARPA and ECDIS with specific reference to the sea stabilised and land display of ARPA vectors. There is reference to collision avoidance but no automatic avoidance mechanism has been included.

It would be desirable for computer simulations of multiple ship encounters to be resolved automatically, where possible. Due to the complex nature of the problem, this will probably improve and evolve over a period of time; given that an unspecified number of ships may approach from a variety of directions, there are an unlimited number of unique combinations. There is no way of recognising that optimal solutions have been achieved, only qualitative evaluation by experience navigators can suggest that the solutions are correct. Once again, for simulation, this solution should reflect the actual practices of seafarers, simulating the reality, rather than necessarily providing an ideal solution.

10.8.6 Close Proximity of Static Dangers and Well-behaved Ships

One of the most advanced collision avoidance and navigation systems is the SPES piloting expert system (Grabowski and Sanborn 1992) developed at Rensselaer Polytechnic Institute. This expert system provides decision support for the navigator for specific ports around the coast of North America, including New York and Valdiz Sound. The expert system includes information about the area being navigated in and approaching ships are monitored. No examples of complex multiple ship collision situations have been discussed in the literature discovered about this system; in the Valdiz example (Grabowski and Sanborn 1995a) the outbound tanker has passed clear before the passenger ship is detected. Fishing and recreational boats are mentioned but no explanation is offered. A more interesting aspect is the ability of the system to determine whether another vessel is passing nearby but remaining within its own traffic lanes. Therefore the SPES does not raise an unwarranted alarm for such a ship, even though it is approaching within the domain. A quoted reasoning cycle time of 15 to 24 seconds is adequate for a single ship but this would be difficult to support in a computer simulation with many target ships. Each SPES node would need to be on a separate computer to obtain adequate performance.

However, this style of navigation in proximity of land would be beneficial to a simulator instructor. It is possible that the track-keeping agent could be expanded to advise the collision avoidance agent of static dangers. The collision avoidance agent might also be able to request danger analysis of an approaching ship. Given the course, speed and ship's position of a target that will infringe the desired domain, the track keeping agent could analyse whether the track of the target ship is complying with a normal navigational route and should pass clear. The track-keeping agent, or even another agent, could be given a ship to monitor to ensure that it maintains a safe track.

10.8.7 Electronic Chart Displays

Electronic Chart Displays are becoming widely available and have been associated with ARPA (Smeaton et al. 1994). The view of ARPA information overlaying chart information, is helpful in planning manoeuvres. Land and sea stabilised information can be compared and rapidly assimilated, reducing information overload.

GPS Satellite navigation can provide extremely accurate navigation. However, navigators are loathe to put total faith in a single piece of equipment, preferring to compare all the available information. Position fixes from land and the depth from the echo sounder form an extremely important part of this equation. Therefore, research is needed on how to provide accurate land and seabed recognition to help provide safe autonomous navigation.

These are needed for safe collision avoidance but will also be beneficial in safe navigation. In fact the two are closely associated, and the problem of avoiding other ships may become easier if the radar images for the land can be identified. Many ARPAs suffer from acquisition of land or other targets preventing acquisition of ships, although Doppler techniques have been used to identify moving targets with some success.

The GIS research at the University of Glamorgan includes research into automatic detection of chart features. This is currently used to amalgamate features from two charts. However, it would be interesting to consider the possibility, whether, knowing the GPS fix, similar comparisons could be performed upon chart and radar data to ensure that the information is all in agreement. MARINES could provide a useful test platform for early work in this area.

There are many other areas of current GIS research that could be applied to future work in the marine environment, such as: the combination of vector and raster chart information; efficient storage and retrieval of chart information; automatic labelling of maps/charts to prevent features being obscured; and the efficient and accurate interpolation of height (or depth) information.

10.8.8 Implementation details

There are several areas in which it is considered that the implementation of MARINES may be improved these are outlined in this section.

10.8.8.1 Windows 32 Bit Platforms

At the time of the inception of the MARINES project Windows 3.1 was widely available. Windows '95 had not been released and the hardware available for the project was not up to the task of running Windows NT. This has now changed, and these, mainly, 32 bit operating systems offer several advantages.

When running a large number of agents simultaneously, one area that has caused some difficulty has been the use of resources. Under Windows 3.1 the resource heaps are restricted to 64K each for the User, Graphics Device Interface (GDI) and System Resources. In particular, the GDI heap is heavily used by the MARINES agents; Device Contexts, Pens, Brushes and Palette colours all take up space. This effectively limits the number of agent applications that can be active at one time. Unfortunately, memory managers, such as Hurricane, rely upon compression of resources for inactive processes; all the agents are actively performing background tasks, preventing this from being an effective solution. Windows 32 bit alternatives, Windows 95 and NT are not tied to such small resource heaps.

Several other factors are also influential in suggesting that a move to a 32 bit environment would be beneficial: processes are run at a lower priority than the operating system, increasing security, a pre-emptive scheduler is used, preventing errant processes from taking excessive time slices; more powerful development environments are now available.

10.8.8.2 Communication Sockets

The MARINES project is based upon DDE connections between the agents. The author is currently an advisor in a short six month project that is presently being conducted to investigate the use of agent technology in Geographical Information Systems (GIS). The research takes many of the ideas from the agent framework in MARINES and builds upon them.

The GIS project is in its infancy, however, one area of research is the use of sockets connections instead of DDE connections. So far, sockets appear to offer a robust solution for an inter-agent communications framework. In addition, sockets have the advantage that the location of the agents is unimportant. The agents may co-exist on the same machine or run on another networked machine.

If this research is successful, then it would be an interesting exercise to scale up the MARINES project to run on several machines, with the agents connected by sockets. This would be helpful when introducing a more sophisticated agent which could run on a separate machine, communicating over a network. A single agent process on a single machine would make compilation and debugging easier during development; the need for processing power would not affect the main environment. If the use of sockets is unsuccessful then, 32 bit, NetDDE may be another solution.

In the future on a distributed system inter-agent communication might also be able to use Agent Oriented Programming techniques, if some of the agents are on separate machines.

10.9 Overall Conclusions

- Automatic collision avoidance has been implemented between the computer generated target ships. This has been shown to operate correctly for most situations involving two power driven vessels in sight of one another. More work is required to tackle some exceptional conditions. For examples, in situations where a rogue ship is involved or the stand on vessel is travelling very fast in comparison to the give way vessel. However, these are the very situations that cause collisions between ships navigated by human navigators (Pike 1997).

  The collision avoidance mechanism is a simplified system that follows the best practices of other researchers in this area. However, the use of several individual agents, each with their own rule base, provides a different perspective. In particular, the ability of the agents to model common misconceptions and characteristics for each individual agent and target ship differs from other research in the field.

- Automatic track keeping has been implemented and shown to maintain the track in a manner similar to a real ship. The track keeping is quite similar to that used in fast time port design systems, but improves upon the level of realism normally available on training simulators for automatic target ship motion.

- Automatic collision avoidance and automatic track keeping have been combined to create a reasonably realistic motion for the computer generated target ships.

  It is the combination of mathematically modelled generic target ship models, collision avoidance, track keeping and a Multi-Agent Systems architecture that makes this work substantially different from the other work in the field. This has been demonstrated to provide realistic target ship motion and to generate the correct course and speed calculations for target ships affected by the current.

  The need for the correct calculations, and the benefits derived, have been discussed in the body of the text; examples of errors in aspect, course and speed have been demonstrated.

- It has been demonstrated that a Multi-Agent System can provide a robust architecture for a simulation, provided that the message passing architecture is suitable.

- The flexibility of the MAS approach has been demonstrated. Agents with different characteristics can be created and linked dynamically to the environment.

- The need for a simulation that includes complex dynamics has been addressed with some success. A number of relatively simple agents complement each other to create a simulation system that behaves in a believable manner. The MARINES simulation takes the computer generated target ships, and hence the simulation, a step towards the complex ship motion found in the real world.

10.10 Have the aims and objectives of the project been achieved ?

- The main aim of this project was to evaluate the benefits of using MASs in marine simulation. This has been achieved for a sub-set of the possible areas where agents could be applied. The benefits of the MAS approach have been described earlier in this chapter, they include, a modular approach, realism, robustness and flexibility. Areas that require further research include the provision of better development tools for MAS.

  Although some of the features could have been achieved using conventional development techniques the complex, realistic, robust simulation is a product of the interaction between the agents and the architecture of the system.

- It has been shown that agents can provide limited target ship manoeuvring automatically. The agents co-operate to perform collision avoidance and track-keeping within a dynamically changing environment.

- The MAS architecture that has been created has provided sufficient communication bandwidth and adequate performance to run up to ten ships and twenty agents, as well as other processes, such as the 3D view. This has been achieved on a single Pentium class PC. Using a hybrid architecture, mixing an object centred autonomous and hierarchical structure, the message framework permits direct communication between the agents, reducing the load on the central environment.

- Agents have been created to perform tasks in the simulation. The agents do perform under both reactive and planned conditions. The overall performance has been shown to be adequate for twenty agents on a Pentium 75 machine. The behaviour has been shown to be realistic for the planned manoeuvres, reactive manoeuvres have been less successful, more research being required for rogue ships.

- A general interface has been created to inter-connect agents; the interface closely resembles the communications in the real world. This interface has been shown to provide the required information using messages based upon requesting, ordering and responding communications.
- The manoeuvring of the target vessels has been judged to closely resemble the manoeuvring of ships in the real world in most situations. This has been achieved both in pre-set test scenarios and under evaluation by simulation and navigation experts. This automatic manoeuvring and adjustment for a changing environment, particularly the individual characteristic manoeuvring of individual ships demonstrates the advantages of the MAS approach.
- The agents send warning messages to the instructor to attract attention and manoeuvre ships automatically to assist the instructor. These features have been judged to provide valuable services for the instructor. However, further research is needed to improve the HCI aspects of the implementation.
- As discussed in section 10.6 the simulation has been shown to withstand some non-critical run-time errors. While this is generally beneficial, it is believed that improvements can be made through the use of a different, and more robust message passing medium.

  Automatic reconfiguration permits the simulation to continue; the remaining agent processes, that were previously advisors to the failed agents, taking direct responsibility for their tasks.
- An MAS has been prepared that permits a number of agent instances to be attached; up to thirty in the present system. Agents with individual characteristics have been run simultaneously. Agents have been started, stopped and replaced at run time. Therefore, the MAS approach does provide a flexible, configurable platform for an instructor station. However, further research is needed to permit the instructor to make good use of this facility.

10.11 Summary

The MARINES project has introduced Multi-Agent Systems techniques to marine simulation. The technology has potential benefits and, in this thesis, the system flexibility and robustness of a Multi-Agent System approach to a marine instructor station have been considered.

The flexibility has been used to provide agents that can dynamically connect to the environment. The agents provided perform collision avoidance and track keeping. An interface has also been created allowing other agents to replace these agents. Limited testing has been undertaken using agents with different characteristics to show how this can alter the manoeuvres and hence the simulation in a realistic manner.

Using the MAS approach it has been demonstrated that the simulation will normally continue to run after some run-time errors that terminate a single agent process. The functionality is reduced but the exercise is not terminated. The ability to continue the simulation in this way has demonstrated the fault tolerance of the MAS approach. This also moves towards a less fragile simulation.

The main thrust of the research has been to improve the motion of the computer generated target ships on micro simulators. These ships are modelled using Newtonian mechanics; in a similar way to that used for own ship models in the past, although the models in MARINES are less sophisticated than those of most own ships. Automatic collision avoidance and automatic track keeping have been provided; the use of agents to perform these tasks has created a complex, dynamically changing simulation. This has assisted in the provision of a number of features: reasonably natural, sea stabilised, target ship motion; the ability for the instructor to alter the exercise dynamically without undue concern for collisions between the target ships; characteristic manoeuvring for a vessel/agent combination; and dynamic response to the changing conditions, such as a different obligation under the rules due to minor changes in the exercise at run-time.

MAS architecture and modelling techniques are rapidly improving. Commercial MASs are beginning to appear and generate considerable interest. Interactions between multiple, relatively simple, agents can produce complex dynamically changing *environments that would be hard to understand and model as a single entity. This has been demonstrated in MARINES taking automatic collision avoidance and track keeping as an example. There is still a strong case for simple exercises on conventional simulators for specific training purposes. However, if exercises are to be created for more advanced training more realistic effects will be demanded; sea stabilised target ship motion and collision avoidance manoeuvres are such effects. The marine training manual STCW(1995) recognises this need, as do Guicharrousse(1990) and Chen (1992).*

Considerably more research is required before the full potential of MAS simulation is realised. MARINES has successfully undertaken an initial voyage of discovery, it returns a sample of the profits and heartache to come. It offers one possible way forward with the enticement that, in the longer term, marine simulation would benefit from robust, adaptable, realistic behaviour.

Jim Moon, 1997

Appendix A : Collision Avoidance

The collision avoidance problem

The collision avoidance problem consists of several stages: detecting vessels in the vicinity; determining whether risk of collision exists; determining the correct action to take; calculating a suitable manoeuvre that provides a suitable passing distance and complies with the action decided upon; ensuring that the manoeuvre does not create a further close quarters situation with another vessel; performing the manoeuvre; monitoring the situation to ensure the manoeuvre is successful; returning to course once the danger has passed.

Detecting vessels in the vicinity

Approaching vessels are detected by sight, sound, radar and any other available means, for example, information from a shore station. The range of detection usually depends upon the visibility, traffic density, etc.

Determining risk of collision

Two techniques are commonly used to determine whether risk of collision exists.

The first technique, defined in the regulations, is to make frequent observations of the compass bearing of an approaching vessel. If the bearing does not appreciably change then risk of collision exists. Even if the bearing changes, risk of collision may still exist if a vessel is close to the observer's ship or is very large.

The second technique is to use a number of observations of the radar bearing and range of a ship, over a period of time, and use trigonometry to determine the course, speed, Closest Point of Approach (CPA) and Time of Closest Point of Approach (TCPA) of the target vessel. Modern Radar sets incorporate Automatic Radar Plotting Aids (ARPA) to perform these functions automatically. ARPAs normally display a vector representing the motion of each target vessel detected; the vectors may be displayed in a number of different ways, showing either the relative motion or the true motion of the approaching ship.

Additionally, in some automatic collision avoidance systems, that use a computer, the most dangerous target is selected using an algorithm based upon the CPA and TCPA (Smeaton and Coenen 1990). At sea, the officer of the watch makes an estimation which may be based upon several additional criteria, including the types of the vessels and their manoeuvrability.

The collision regulations

If risk of collision exists then the collision avoidance manoeuvres should follow the International Regulations For Preventing Collisions at Sea (IRPCS 1989). The regulations determine the responsibilities between vessels for the majority of situations involving two vessels. These responsibilities normally nominate a stand on vessel and a give way vessel; the action for the give way vessel is also constrained in general terms, preventing dangerous manoeuvres.

The determination of the give way and stand on vessels depends upon a number of descriptors such as the relative positions of the two ships and the visual aspect of one to the other.

Taking action

The action determined from the rules is generic rather than specific. I.e. an alteration to starboard, should not alter to port, a bold alteration, etc. The actual alteration to make has to be determined either by trigonometry or using experience and judgement. Knowing the course and speed of the two vessels it is relatively easy to determine possible alterations of course that will result in the desired passing distance; assuming that the target maintains its present course and speed, and solutions are possible. There are in fact four possible solutions to the trigonometrical problem, although some of these may contravene the regulations.

Trial Manoeuvre

Before making a manoeuvre it is wise to ensure that the alteration will not cause a close quarters situation to develop with other traffic. Once again trigonometry is used to perform a trial manoeuvre to determine how your action will affect other traffic. On a real ship ARPAs have the ability to display the effects of a trial manoeuvre.

Performing the manoeuvre

Having determined a suitable manoeuvre, the navigator will alter course or speed accordingly. Normally, an alteration of course is preferred, being more immediately apparent to the other vessels and easier for the navigator to effect. Large slow and medium speed diesel engines used aboard merchant ships are normally run at a steady speed, on heavy grade oil, except when manoeuvring in port. Such machines do not respond well to rapid changes of power. Even an alteration of course takes some time on a large vessel, particularly as the navigator will not wish to apply large helm angles, unless they are really necessary.

Monitoring the situation

A manoeuvre will seldom result in precisely the planned passing distance. Delays in executing the manoeuvre, inexact turning circles, manoeuvres by the other ships, etc., will all conspire to prevent this. Therefore, the navigator has to monitor the developing situation carefully.

Returning to course

An navigator will often use previous experience as a guide in determining when to return to course. In general this will occur as soon as possible after the danger has passed. If the ship is on relatively short, coastal, courses then the navigator will also apply a correction to return the ship to the desired track relatively quickly. On lengthy ocean passages corrections are made less frequently as deviations of a mile or two are of little consequence, it being more important to travel by as short a route as possible.

If the traffic density is high then a trial manoeuvre may be necessary before returning to course, in the same way as before the manoeuvre.

The MARINES collision avoidance agents

The agents perform collision avoidance in much the same way as that outlined above. Each agent sends regular messages requesting all the visible targets and stores the results. At intervals, the range, bearing and aspect of these approaching targets is used to calculate the CPA, TCPA, course and speed of the visible targets, and determine the most dangerous target. If a dangerous target is very close then the reactive part of the agent takes evasive action, otherwise the rule base is consulted in order choose the general action to take. For example, alter course to starboard. The actual alteration required is then calculated and a trial manoeuvre assessed. If this is successful, the auto pilot is set to the desired course for the manoeuvre, and the ship begins to alter course. The agent continues assessing the situation, ready to take further action, if required. Every so often, the agent will try a trial manoeuvre to return to course, and when this is accepted it will set the auto pilot accordingly.

The track keeping problem

Basic track keeping without a current is a two stage problem. The first stage is maintaining a straight track, applying a correction to return the ship to track. The second stage when approaching a way point is to decide when to commence the turn, this depends upon the manoeuvring characteristics of the ship, the speed that the ship is travelling and how far off the track the ship is.

To determine the current that is affecting the ship historical position data has to be stored. This is used to infer the set and drift that is being experienced. The course that the ship has to steer to counteract the current can then be calculated.

Maintaining a straight track

Maintaining a straight track without current applied is relatively simple. The distance off the track is calculated and a correction is applied that returns the ship to the track. The size of the correction varies according to how far off the track the ship is. The correction is normally applied in increments of whole degrees, at intervals a few minutes apart, rather than as a continuously varying correction. One reason for this is that frequent changes of course affect the accuracy of the information inferred from historical data. It would therefore become difficult to determine the motion of target ships in the area and the set and drift of a current, if the course was continuously altered.

Determining when to turn

The navigator has to decide when to begin a turn before arriving at the waypoint. The actual distance is a function of the vessels' turning circle at the speed and rudder angle that will be used. There is also a latency which results in additional forward motion when entering the turn that has to be allowed for. The latency is due to a number of factors including the time taken from putting the helm over until the rudder reaches the specified angle and the inertia of the vessel.

The navigator of a merchant ship will not usually calculate these factors for each turn. More frequently the navigator applies previous experience and a knowledge of the manoeuvring characteristics of the ship during performance trials[1].

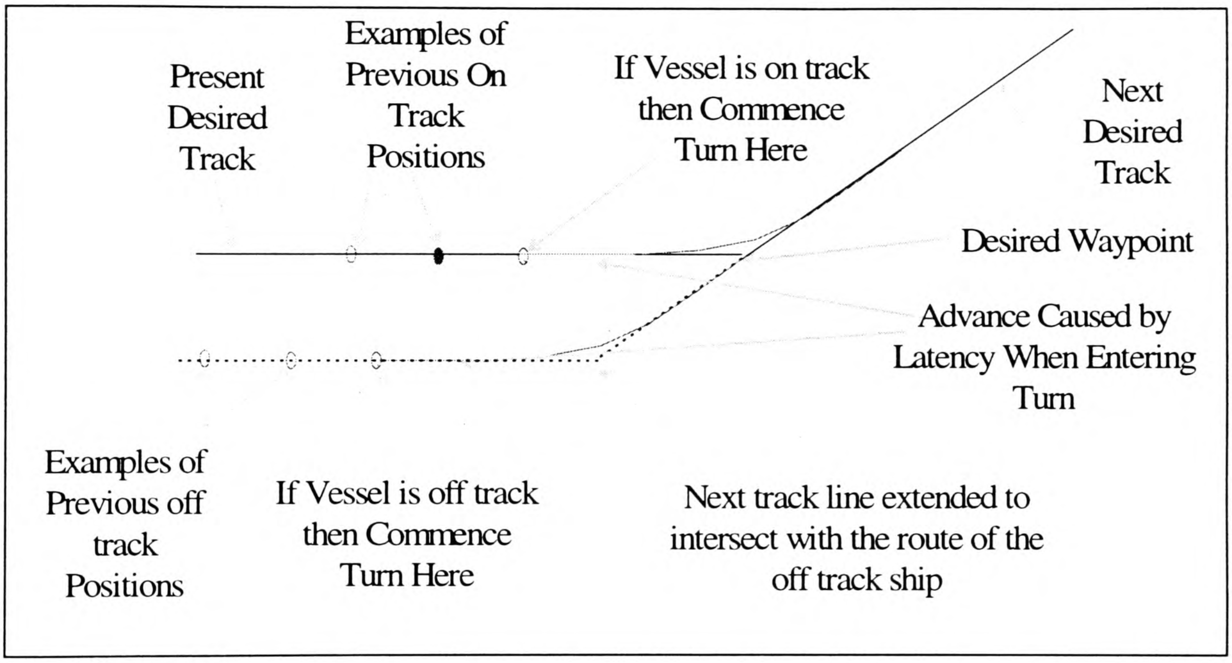


Figure A-1 Deciding when to commence a turn at a waypoint

Figure A-1 shows a typical alteration at a waypoint. If the ship is precisely on track then deciding when to turn can be based upon the distance to the waypoint. When the distance is equal to the expected advance caused by latency plus the relevant turning distance then the helm is put over and the ship gradually alters course onto the next

[1] When a new ship is delivered performance trials are performed to ensure that the manoeuvring characteristics approximately match the specified design performance. They are performed at particular displacements and speeds; the actual characteristics in service may differ according to the trim, stability, speed, displacement, etc. of the vessel.

track. If the ship is originally off track then the turn may have to commenced earlier or later according to the direction of the next track, an example of this is also shown in Figure A-1.

Counteracting the current

The set and drift of the current that is likely to exist in a particular area can be obtained from current atlases and charts. The true set and drift of the current being experienced is estimated by comparing the expected dead reckoning position of the ship with the actual position the ship is in.

Once the set and drift of the current have been estimated then, knowing the ships speed, the course to steer to maintain a desired track may then be calculated.

The MARINES track keeping agents

Once again the track keeping agents closely follow real practice. At present the simulated agents have two advantages over the navigators in the real world. Firstly, the positions from their simulated satellite navigation systems are extremely accurate, making the estimation of the current easier. Secondly, the computer estimation of the time to turn can be considerably more accurate than the normal estimates. These advantages are offset by the agent having to use the simulated auto-pilot to effect the turns.

In MARINES the estimation of when to turn is made by applying the agents beliefs of the vessels manoeuvring to a formula. Figure A-2 shows how the allowance that must be made for the turning circle of the ship is determined. The actual value will depend upon the size of the alteration of course. In any case, if the agent's belief of the turning circle is wrong then the vessel will turn early or late.

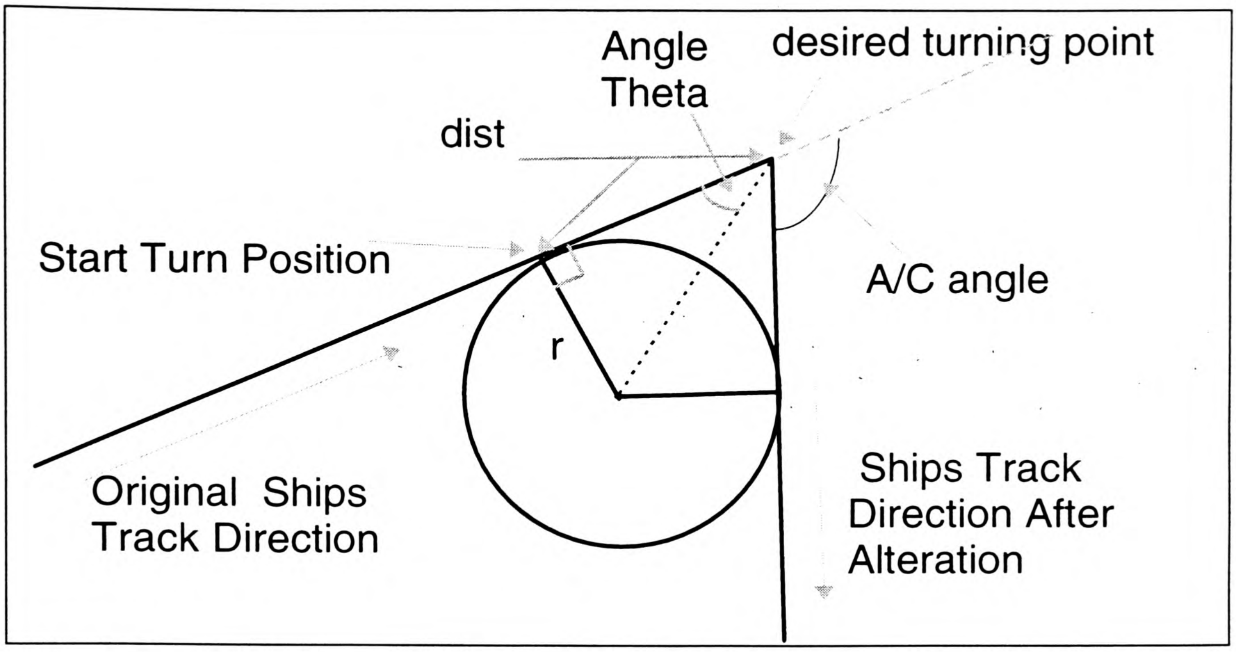


Figure A-2 Allowing for the turning radius at a waypoint

In Figure A-2 :

dist = the distance between the start turn position and the desired turning point.

NOTE: the desired turning point is a waypoint if the ship is on track, as shown in Figure A-1.

r = the vessel's radius of turn

A/CAngle = The size of the angle of alteration

From the diagram it can be seen that :

Theta = 180 - A/CAngle / 2

And:

Tan(Theta) = r /Dist

Therefore:

dist = r / Tan (180 - A/CAngle / 2)

Of course, the advance due to latency also has to be added to 'dist', in order to determine when to commence the turn.

If a collision avoidance agent is attached to the ship then the course to steer is then passed to the collision avoidance agent for processing. Otherwise the agent requests the course directly.

Appendix B : The Collision Avoidance Rule System

Introduction

The production rules have been chosen as an example of the technical development strategy used for the agents in MARINES. Production rules have been chosen as several previous collision avoidance systems have shown them to provide acceptable results. The rules are also relatively easy to understand without needing to consult the programming language.

The production rules are written using a text editor and then compiled into a symbolic machine format before being used in the MARINES agents. During this compilation, source code files are generated for inclusion in the agent program that specify the legal rule conditions. These source files also provide functions for initialisation and retrieval of the conditions used in the rules.

At run time the symbolic production rules are parsed in a top down, forward chaining manner until a rule fires. That is, each rule is tested from the start of the file until one is found in which all the conditions are true, or the end of the rule set is detected.

Rule Grammar

The grammar for the rules is very straightforward. A simple rule compiler is used to convert the rules from normal ASCII text into a symbolic form. The compilation is done for two reasons, firstly, the parse of the symbolic representation is more efficient and, secondly, the syntax of the rules is checked before the rules are used in the main program, reducing the danger of an error at run-time.

An example rule will be :

IF Condition1 AND Condition2 AND... THEN Action1 RULEEND

The BNF of the rules grammar

Rules = <Rule> [Rule] <RULESDONE>

Rule = <IF> <Condition> [<AND> <Condition>] <THEN> <Action> <RULEEND>

Condition = <id>{FROM Set of conditions}

Action = <id>{FROM Set of Actions}

id = <Alpha>[<Alpha>|<Numeric>]<Space>

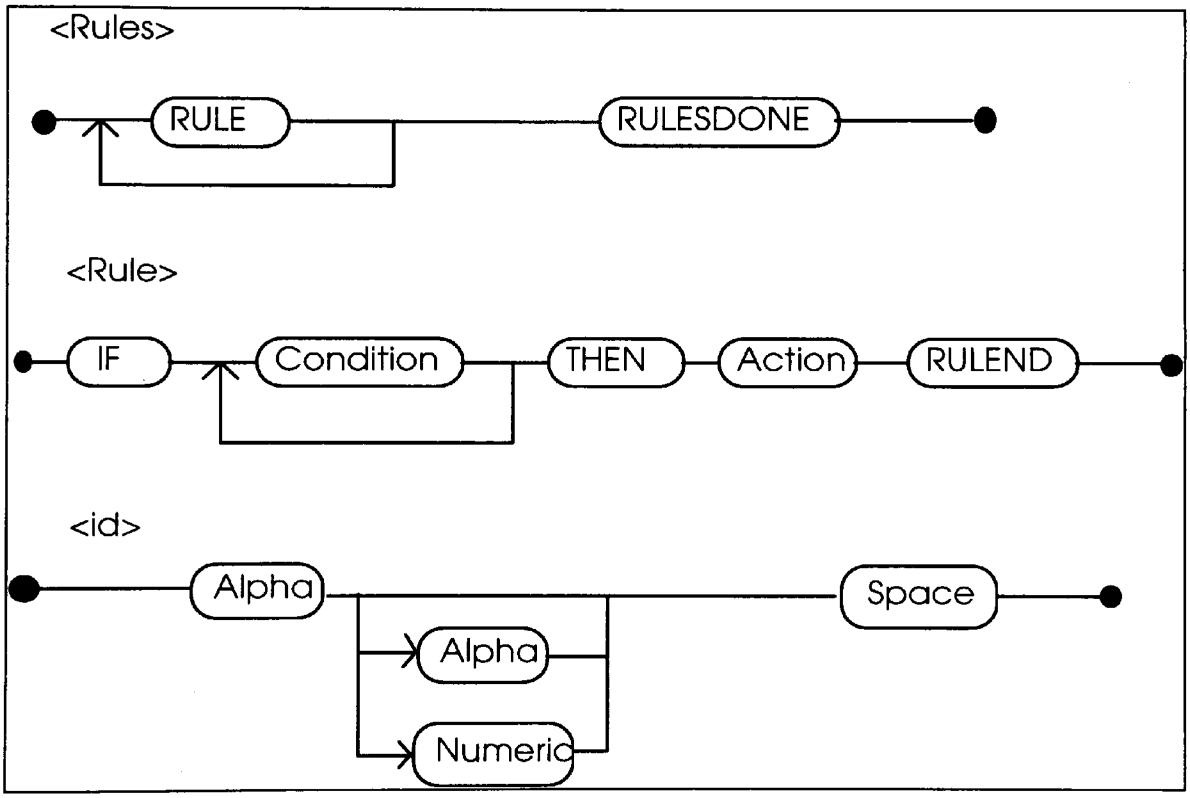


Figure B-1 The rules syntax in diagrammatic form

Compiled rule data base format

The compiled rules are stored in the format shown in Figure B-2. A pointer is held to the start of each rule. For each rule, the condition symbols are stored in order, followed by a THEN symbol and an action. At run time the first pointer is used to access the first rule, if a condition is FALSE then the next pointer is used to access the second rule, and so forth. If an ACTION is reached then the parse is terminated and the ACTION passed back to the program for processing.

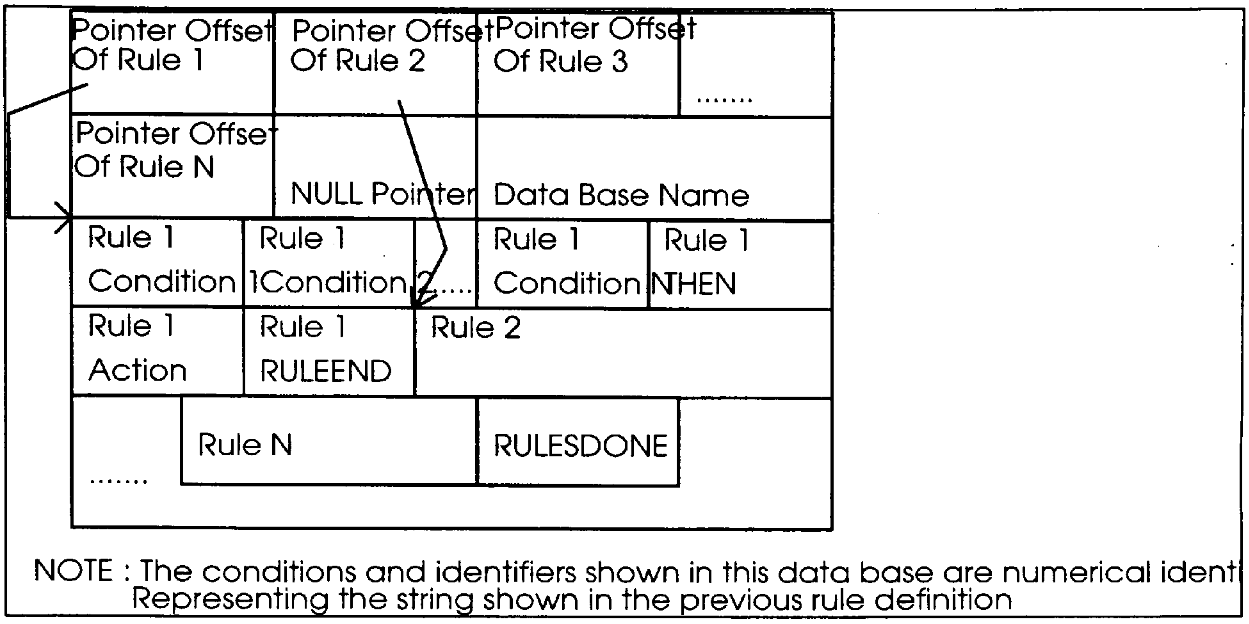


Figure B-2 The compiled data base format

Determining the rules

In order to determine the rule set to use the collision regulations (IRPCS 1989) and seagoing experience were considered. A tree showing the probable priority for manoeuvres was drawn up as shown in Figure B-3.

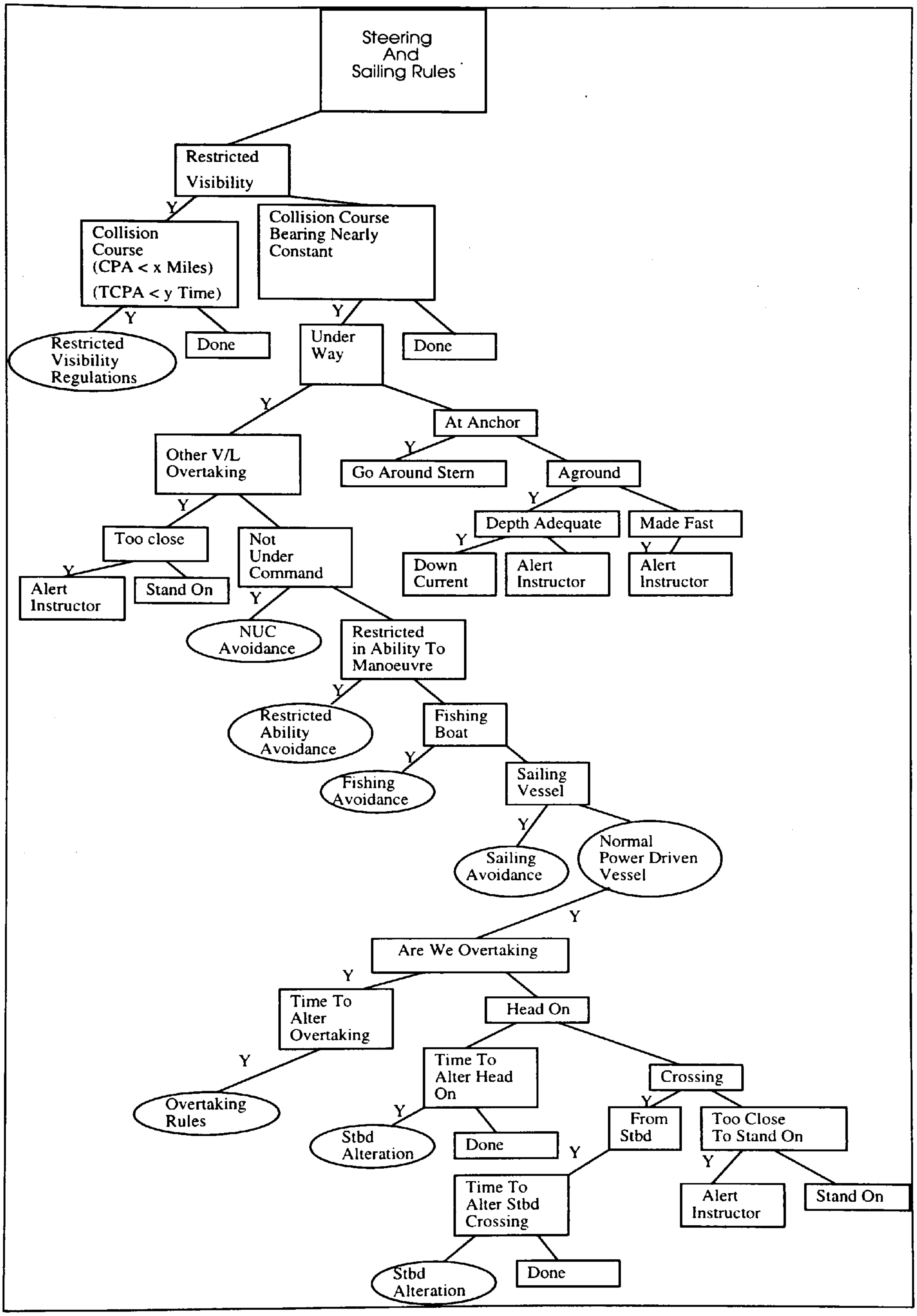


Figure B-3 A first attempt at organising the rules

Next, a rule set was designed for power driven vessels in sight of one another. Special circumstances such as restricted visibility, sailing vessels, etc. were not taken into account. Even with this simple avoidance in mind some rules had to be re-ordered after some initial experiments.

The format of the rules

The source text format of the rules is shown in this section.

Comment lines begin with a '/' and are ignored by the compiler. Therefore, each rule statement must begin on a line without a comment. Other than this the format is reasonably free form.

Each rule set begins with a header naming the rule set. This should be the same as the body of the file name that the rule set is using. In this example the source rule set will be called "PDVRules.Dat" and the compiled rule set will be called "PDVRules.Rul".

```
/ REM Power driven vessel rules
/    Version 1.01
/    Jim Moon
RULESNAME PDVRules
```

A new rule set can be called from the rule set in use if a particular fact is asserted. For example, if a trawler target is detected then "TrawRule.Rul" can be loaded.

```
/ If you are not dealing with a normal Power Driven Vessel which
/ is under way Choose Another Rule Base
IF TgtTrawler THEN
 NewRules TrawRule
RULEEND
```

A typical rule will consist of a number of conditions and an action. A response function has to be encoded into the Agent to perform a relevant action. The following example shows an overtaking situation, If the agent is overtaking a ship that is steering the same course, nearly dead ahead of the agents' own ship then it makes an alteration of course to starboard.

/ Deal with Agent overtaking situation No 10
IF SameCourse AND OwnOverTaking AND TgtAheadOfOwn THEN
StbdAlteration
RULEEND

The Rule Compiler

Once the rule set has been designed a list of the condition, action and macro <ids> has to be created. This is used by the rule compiler to ensure the syntax of the rule set is correct.

The rule compiler is only used for converting the rule set before the agent software is built. In this prototype version the implementation has been created as quickly as possible; the user interface is utilitarian and no effort has been made to optimise the compilation. In any case, the relatively short rule sets take only seconds to compile.

The compiler consists of a scanner and parser. As the parse is performed the compiled rule set is stored in memory. If an error is found the line and place in the source rule set is displayed on the screen. Once all errors have been eliminated the symbolic version of the rule set is written out to disk. All the newly created files have the same filename body as the source rule set file, each has a different extension according to what it contains; '.dat' for the source rule set, '.rul' for symbolic rule set, '.h' for the header file and '.cpp' for the code file.

The Scanner

The scanner steps through the source file, skipping comment lines and obtaining each reserved word and <id> in turn. Once obtained, each <id> is compared with the list of previously declared <ids> to ensure that it is legal.

Initially, it was found that the task of preparing the list of <ids> and, later, adding them to the agent program was somewhat onerous and prone to error. Therefore, functionality was added to the scanner to produce the necessary files for inclusion in the agent program automatically.

The Parser

The parser ensures that the layout of the rules adheres to the grammar. The parser also stores the compiled rules as they are created and, if the compilation is successful, writes the newly created symbolic rule set to disk.

The space required for the compilation is allocated as an array in this simple parser, this was considered adequate for this prototype version.

Concluding remarks

The design of the rule system for agents in MARINES has several advantages in the creation of agents for simulation:the performance at run-time is good, each symbol is a held as a single integer each rule is parsed only until a condition fails, the rules are only parsed until one fires; storage of the compiled rule sets at run-time requires little space; the rules are checked for syntax before run-time, reducing the possibility of errors in the simulation; the code for manipulating the symbols is generated automatically for inclusion in the agent project.

To improve the ease of use of the rule sets and inference engine, further research into creating some common agent components is required. It should be possible to create response functions in the rule editor for the actions returned from parsing the rule sets. Interpreted frames, used in some agent implementations, provide a possible solution, however the resulting performance is questionable for a continuous simulation. Automatic generation of 'C' and 'C++' macros or function prototypes may also provide a solution. Furthermore, an object frame work could be created that provides standard response functions, it being left to the agent creator to fill out new functions, in a similar manner to that used in Microsoft Visual C++ (Kruglinski 1995).

Appendix C : Sample Questionnaire

MARINES Evaluation

As with many computer products simulators are becoming cheaper. The multi million pound simulation centres with full time professional support staff are still being built. However, a less costly, breed of simulator is gaining popularity, particularly in the smaller, privately owned marine colleges. These simulators are often based around micro computers, e.g. Maritime Dynamics, Transas, PC Maritime, etc. The instructors for these simulators may only run a simulation as a small part of their daily tasks and can find the task quite daunting. Yang Hong Chen studied the use of marine simulators in the far east and found that the instructors often felt intimidated in having to control the simulation and simultaneously deal with the students. A further area of concern is that of providing realistic simulation exercises, this is to some extent exacerbated by the need to provide exercises that are easy for the instructor to control. In order to make the target ships easy to control their manoeuvring characteristics may be enhanced and/or their manoeuvres constrained in some way. E.g. the target ships may not be affected by the set and drift of a current and they may follow a pre-defined track without deviating for dynamic changes in the environment.

Considerable research has been performed on intelligent agents in dynamically changing situations where they have been shown to perform effectively, e.g. Tileworld. Furthermore, Multi Agent Systems(MAS), where several co-operative agents work together to solve problems, have been analysed using simulations, for example, the Phoenix project. MARINES is a test bed for evaluating the use of an MAS as a means of providing assistance to the instructor of a marine simulator and it is hoped that the results will be valuable in other areas of simulation. The approach also takes into account research on automatic collision avoidance and track keeping. However, the MARINES testbed is only a research vehicle and as such is not designed to fully exploit these capabilities, rather it is intended to demonstrate the potential and determine whether they are valuable.

This questionnaire is designed to consider the benefits and constraints imposed by the use of an MAS in a simulator. Initially, in section one, a few questions are set to discover the respondents area of interest, in section two there are several questions about the merits of the features that are being considered, the third section is to determine whether, in the respondents opinion, an MAS is helpful in providing these features. Finally, your opinion is sought as to the use of MAS for other features that have not yet been implemented.

Feel free to add comments or suggest amendments to the questions.

Section 1

Q1 Please tick all answers that apply.

Are you :

a lecturer in maritime studies ? __________

a lecturer in Artificial Intelligence __________

a simulator developer ? __________

an experienced simulator instructor ? __________

an occasional simulator instructor ? __________

an MAS developer ? __________

an MAS researcher ? __________

a ship's officer ? __________

a student in maritime studies ? __________

Other (please specify) ______________________________

Q2 Please tick all answers that apply.

Do you use a simulator for:

Student training __________

Port design __________

Testing ship models __________

As a marine student __________

Research (please specify)______________________________

Other (please specify)______________________________

Q3 Please tick all answers that apply.

Do you design simulator exercises for:

Student training __________

Port design __________

Testing ship models __________

Research (please specify)______________________________

Other (please specify)______________________________

Q4 Please tick the first answer that applies.

Do you consider yourself :

- a fluent computer programmer ________
- a capable computer programmer ________
- a fluent computer application user ________
- a capable computer application user ________
- an infrequent computer user ________

Q5 Please tick the first answer that applies.

Do you use a simulator :

- Daily ________
- Once a week ________
- Once a month ________
- Occasionally ________
- Less than once a year ________

Q6 Do you have specific interests in MAS systems ? ________

If so, please specify

____

____

____

____

Answering the remainder of the questionnaire.

Please select the answer that most accurately reflects your view.

If you are unable to select an answer place a cross in the "Unable to Comment box". If the reason is a poorly phrased or unnecessary question , please indicate the problem in the comments section. Move on to the next question.

Otherwise place a cross in the box that reflects the level of certainty with which you have selected the answer.

E.g. if you are sure that that you have understood the question and certain that only one of the given answers is applicable, put a cross in the Certain box:

Certain ☒ Confident ❑ Open-Minded ❑ Unable to comment ❑

and insert the letter in the answer letter space.

answer letter C

Then show how important you feel the feature is in creating a good simulation by putting a cross in the Important box.

Crucial ❑ Important ☒ Worth-while❑ Unimportant ❑

Due to the complex nature of this evaluation a section is then included for any comments you would like to make.

E.g.

Comments I believe that ease of control allows more demanding exercises to be developed and that the student is unable to detect the enhanced manoeuvring.

Section 2

Q1 Do you believe that computer generated target ships should :

a) be modelled to take into account the characteristics of a specific ship.

e.g. if the 'M.V. Riverville' is shown on the student screen the mathematical manoeuvring model should match that ship as precisely as possible, disregarding the additional interaction this requires from the instructor.

b) be modelled to take into account the different ship types.

e.g. if a fishing boat is shown on the student screen the mathematical manoeuvring model should approximately match that generic ship type.

c) be easy for the instructor to control, despite accelerating, turning and stopping in an unrealistic manner.

e.g. the target ship models are rudimentary and turn, accelerate and stop almost immediately.

answer letter ________

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Q2 Do you feel that the computer generated target ships should :

a) be unaffected by the set and drift of a current.

b) immediately alter course in response to a change in the current without deviating from the pre-planned route.

c) gradually alter course in response to a change in the current without deviating from the pre-planned route.

d) allow themselves to be pushed off track by the current and respond to the off track position by applying set to the course in order to return to track. The targets should always display the correct visual 'aspect' allowing for wind and current.

answer letter ________

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Q3 Ideally, when computer generated target ships meet other computer generated target ships they should:

a) automatically manoeuvre observing the collision regulations for the type of ships involved.

b) Manoeuvre in some pre-defined manner.

c) Never meet due to the exercise preparation.

d) Be carefully monitored by the instructor, who intervenes and manoeuvres the vessels.

e) Alert the instructor to the danger, who then intervenes and manoeuvres the vessels.

answer letter ________

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Q4 Ideally, when computer generated target ships meet a ship controlled by students they should:

a) automatically manoeuvre in a natural way, observing the collision regulations for the type of ships involved.

b) manoeuvre in some pre-defined manner.

c) be carefully monitored by the instructor, who intervenes and manoeuvres the vessels.

d) alert the instructor to the danger, who then intervenes and manoeuvres the vessels.

e) never manoeuvre as the exercises are designed to force the students into action.

answer letter ________

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Q5 Ideally, do you believe that computer generated target ships should :

a) follow pre-planned routes free from run-time intervention, turning at waypoints without responding to dynamic changes in the simulation.

b) follow pre-planned routes, without deviation, the course and speed may be over-ridden by the simulator instructor taking control of a target ship.

c) be semi-autonomous, attempting to follow planned routes, but deviating to avoid other traffic and alerting the instructor to danger. The automation may be over-ridden by the simulator instructor taking control of a target ship.

d) be fully autonomous, attempting to follow planned routes, but deviating to avoid other traffic.

answer letter ________

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Section 3

Q1 MARINES provides limited collision avoidance between target ships. In your opinion :

a) this enhances the realism of the simulation.

b) could be worth-while if, and only if, complete collision avoidance was provided.

c) despite instructor over-ride, it removes too much control from the instructor.

d) prevents the students' ship from becoming involved in close quarters situations.

answer letter ________

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Q2 MARINES provides limited track keeping for target ships. In your opinion :

a) this is a valuable feature.

b) could be valuable if, and only if, complete track keeping and land avoidance was provided.

c) despite instructor over-ride, it removes too much control from the instructor.

d) it is less useful that the earlier less dynamic pre-planned routes.

answer letter ________

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Q3 The combination of track keeping and collision avoidance together provide features that one or the other, on its own would lack. E.g. the ability to respond dynamically to a changing simulation. In your opinion:

a) is this a valuable feature.

b) could it be valuable if, and only if, complete collision avoidance, track keeping and land avoidance were all provided.

c) is it too complex for the instructor to control.

answer letter ________

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Q4 Each agent in the MARINES system is a separate program. It can be shown that in most cases a run-time software error in one agent will not bring down the whole software system, only that agent process.

An error that crashes the Windows Kernel, User, GDI or DDEml run time libraries can, however, lock the system. This applies in any case to any Windows 3.1, 16 bit application. These problems are, however, implementation specific. Windows NT, for example, provides greater protection to prevent interaction between processes, reducing the danger of such an error.

An error within the main MARINES instructor station which forms the hub for the communications will also terminate the simulation.

However, moving some of the functionality from the instructor station into the agents reduces the complexity, the number of lines of source code and hence the danger of instructor station failures due to a software bug. Maintenance of the actual Instructor station code is also reduced, many new improvements being possible within the agent modules.

New agents can be developed, not only by the simulator manufacturer, but also by third party vendors, allowing new features to be added without touching the original source code, provided that a suitable interface has been included in the original instructor station.

In your opinion:

is this a worth-while feature?

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Section 4

Q1 In your opinion does the head on collision scenario without current accurately depict a meeting between two ships ?

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Q2 In your opinion does the head on collision scenario with a northerly current accurately depict a meeting between two ships ?

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Q3 In your opinion does the head on collision scenario with a southerly current accurately depict a meeting between two ships ?

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Q4 In your opinion does the crossing collision scenario without current accurately depict a meeting between two ships ?

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Q5 In your opinion does the crossing collision scenario with a westerly current accurately depict a meeting between two ships ?

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Q6 In your opinion does the overtaking collision scenario without a current accurately depict a meeting between two ships ?

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Q7 In your opinion does the overtaking collision scenario with a northerly current accurately depict a meeting between two ships ?

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Section 5

Q1 In an MAS it is possible to launch and stop agents dynamically. This allows an agent with different characteristics to be put in control of a ship at run-time. E.g. a more sophisticated agent could simulate a pilot 'taking the con' for a river passage or the captain taking over from an inexperienced officer :

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Q2 A future area that may be worth researching is the provision of tugs. It may be possible to create agents to navigate tugs. The tug captain agents would receive instructions to meet a ship at a specific place, they would then plan how to reach the spot, or possibly have a pre-planned route to follow.

When taking the tugs lines the tug captain agent would control the tug and another agent determine if it is possible to pass a line. The agents could be programmed to consider the loss of stability and other factors relating to their own safety. Thus, it may be possible to provide highly realistic interactions between several tug operation simultaneously.

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Q3 An MAS may be capable of providing realistic fleet manoeuvres. E.g. an agent may be provided to keep station with another ship. Several of these agents could operate to perform as a fleet. The agents would still be capable of independent action. E.g. trying to avoid attack or planning a new strategy if the ship is damaged and unable to maintain formation.

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Q4 A car simulation has been used to investigate MASs. Conversely, it is possible that an agent could act as the driver of a car in a simulation. Using this paradigm, further research could consider whether an agent driven car could successfully overtake and avoid other cars, producing a more realistic simulation. Agents could then be provided that simulate reckless, over-cautious, drunk drivers etc.

Comments:____

____

____

Certain ❑ Confident ❑ Open-Minded ❑ Unable to comment ❑

Crucial ❑ Important ❑ Worth-while ❑ Unimportant ❑

Appendix D : Glossary of Terms

Overview

This appendix contains a glossary of some of the terms used in this thesis, that are not in everyday use. The terms are explained only in the context of this thesis, although, in some cases, they may apply elsewhere in different contexts. Terms that are defined in the glossary appear in **Bold Type** *if they are used in the description of another term. For Example :*

ARPA An acronym for **Automatic Radar Plotting Aid**.

In this example **Automatic Radar Plotting Aid** *is itself defined elsewhere in the glossary.*

Terms and Definitions

Agent [1]One who acts (on behalf of).

(see 3.2.2) [2]Something that produces an effect.

[3]The definitions above have combined to loosely define a software process, or robot, where several simple, autonomous, agents, acting on each others' and/or the user's behalf, create a complex effect. In Multi-Agent Systems the agents are usually intelligent, autonomous, co-operative, communicating software processes.

AOP An acronym for **Agent Oriented Programming.**

Agent Oriented Programming (AOP) this is a development of the idea of [2]**Object Oriented Programming**. The actors or modules become agents in **AOP** and they have mental states consisting of beliefs, capabilities and decisions.

AI An acronym for **Artificial Intelligence.**

Artificial

Intelligence(AI) The design and implementation of computer programs that can emulate human cognitive ability. **AI** tries to solve problems without a precise mathematical or sequential recipe for their solution.

Aspect The aspect of a vessel describes the way that an approaching ship appears to an observer. It is actually a measure of the angle that the observer subtends relative to the heading line of the approaching ship. For vessels in sight of one another, where risk of collision exists, 'aspect' is an essential indicator when determining whether the observing vessel should give way or stand on.

ARPA An acronym for **Automatic Radar Plotting Aid**.

Automatic Radar Plotting Aid An addition to a marine radar that produces vectors on the radar screen that assist the navigator in determining the speed, course, CPA and TCPA of an approaching target. Most modern ARPAs form an integral part of the radar itself, and the term then refers to the complete radar device.

C A block structured, 3rd generation, computer programming language. This is a medium level language that has less strict type checking than, for example, PASCAL.

C++ Based upon **C, C++** is a superset of the **C** language that adds **Object Oriented** extensions. Powerful programming concepts such as **Encapsulation**, **Inheritance** and **Polymorphism** are relatively easily achieved. **C++** also supports stronger type checking than **C**.

Chart A marine map.

CPA An acronym for **Closest Point of Approach**.

Closest Point of

Approach (CPA) The minimum distance that it is calculated that a radar target will pass, if both the observer's ship and the target maintain course and speed.

DAI An acronym for **Distributed Artificial Intelligence.**

DDE An acronym for **Dynamic Data Exchange**.

Distributed Artificial Intelligence(DAI) The study of the application of **Artificial Intelligence** intelligence techniques to distributed computer systems. Blackboard Systems and **Multi-Agent Systems(MAS)** are often considered sub-fields of **DAI,** however, **MAS** research sometimes exceeds the boundaries of **DAI .**

Dynamic Data Exchange(DDE) A low level message passing system that is an integral part of the Microsoft Windows 3.1 Operating Environment. DDE permits blocks of data to be passed between processes, at run time, using a Client/Server architecture.

ECDIS An acronym for **Electronic Chart Display.**

Electronic Chart Display A computer system that displays a marine chart in an electronic form. It is used on the bridge of a ship as a replacement for the paper charts. Some simulators also have **ECDIS** terminals.

Encapsulation In a programming language, the wrapping of functionality and data together. This assists in the creation of multiple instances of objects. (See also **Object Orientation, Object Oriented Programming and C++**)

Expert System A computer based advisory system that encompasses domain knowledge about a specialised subject. Such systems are usually composed of a **knowledge base** and an **inference engine.**

Inference Engine An **inference engine** performs the reasoning for an **expert system.** This is usually achieved by parsing a set of **production**

rules and comparing the conditions to the knowledge stored in the **knowledge base.**

Inheritance In a programming language, the ability to reuse and build upon the functionality of an existing object. This assists in the creation of **polymorphic** objects. (See also **Encapsulation, Object Orientation, Object Oriented Programming and C++**)

Instructor Station The area used by a simulator instructor to set up and control exercises performed on the simulator. This usually includes one, or more, computer terminals, simulated VHF radio sets and internal communications panels.

Knowledge Base A store of domain specific knowledge. This knowledge is used by the reasoning **inference engine** of an **expert system.**

MAS An acronym for **Multi-Agent System**.

Multi-Agent System (MAS) A Multi-Agent System (MAS) is a collection of relatively simple **agents**. A MAS tries to produce complex effects through the use of a large number of relatively simple, co-operating agents. Each agent is able to act autonomously to achieve goals, without continuous user intervention.

Object Orientation A software development paradigm based upon information hiding, message passing and abstraction of design from real world objects.

Object Oriented Programming $_1$A method of producing computer programs from Object Oriented Designs. Certain computer languages are designed to specifically support **Object Oriented Programming** constructs. Examples are the Smalltalk and Eiffel languages.

$_2$OOP views a computational system as a connected set of modules or actors that communicate via messages (Shoham

1993). The messages are more like a language than the machine specific messages that are normally passed in the first definition.

Polymorphism Meaning many shapes, in a programming language or design, this is the ability to reuse most of the common functionality of an existing object through **inheritance**, and alter the way some of the functionality operates to create a new relative of the original object. A much cited example in computer graphics is that plotting a point on the screen consists of two primitive operations Move and Draw. Plotting a circle also has two primitive operations, Move and Draw. The Move operation is common to both and can be inherited, the Draw operation has to be different for each, to draw a point or a circle. They are both derived from a class of graphic objects, giving a taxonomy of graphic objects containing many shapes. (See also **Encapsulation, Object Orientation, Object Oriented Programming and C++**)

Port $_1$The left hand side of a ship when looking forward towards the bow.

$_2$ A place where ships are loaded and unloaded.

Production Rules Rules that are parsed by an **inference engine**. They are of the general form :

IF condition AND Condition THEN

Action

Power-Driven Vessel "The term "power-driven vessel" means any **vessel** propelled by machinery" (IRPCS 1989)

Starboard (Stb'd) $_1$The right hand side of a ship when looking forward towards the bow.

TCPA An acronym for **Time of Closest Point of Approach**.

Time of Closest Point of

Approach (TCPA) The length of time that it will be before the calculated **CPA** is reached.

Track-keeping The task of making a vessel follow a desired track. On a real ship the desired track is usually planned in advance and plotted on a marine **chart** in pencil. The ship may be pushed off the track by natural forces or have to deviate to avoid an unplanned obstacle. The heading of the ship has to be adjusted to return the ship to the desired track.

Vessel "The word "vessel" includes every description of water craft, including non-displacement craft and seaplanes, used or capable of being used as a means of transportation on water" (IRPCS 1989)

Appendix E : References

Adams S.S. and Nabi A.K., 1989, "Neural Agents - A Frame of Mind", *OOPSLA '89 Proceedings*, October, p139~149.

Addanki S., 1987, "Connectionism", *Encyclopaedia of Artificial Intelligence*, Volume 1, John Wiley and Sons.

Agre P.E. and Chapman D., 1987, "Pengi: An Implementation of a Theory of Activity", *Proceedings of the 6th National Conference on AI*, p268~272.

Alpaydin E., 1993, "Multiple Neural Experts for Improved Decision Making", *The Biology and Technology of Autonomous Agents*, Steels L. (Ed), Springer-Verlag, p451~458.

Bird S.D., 1993, "Toward a Taxonomy of Multi-Agent Systems", *The International Journal of Man-Machine Studies*, Vol. 39, p689~704.

Blackwell G.K., Colley B.A. and Stockel C.T., 1988, "An Intelligent Knowledge Based System for Marine Collision Avoidance", *Proceedings of the Conference on Maritime Communication and Control*, Institute of Marine Engineers, Paper 19, (Oct.), p167~174.

Blackwell G.K. and Stockel C.T., 1989, Simulation of Ship Encounters Using Multiple Linked Intelligent Systems", *Proceedings of the Summer Computer Simulation Conference, SCSC '89*, p766~771

Blackwell G.K. and Stockel C.T., 1990, "Optimal Strategies for a Marine Real-Time Expert System", *Proceedings of the European Simulation Multi-Conference*, p294~299.

Blackwell G. K., Rangachari J. and Stockel C. T., 1991. "An Intelligent Interactive Environment for A Maritime Real-Time Expert System", *Proceedings of the Summer Computer Simulation Conference*, Baltimore, USA, (Jul.), p358~363.

Brennen J. E., Walenty M. E. and Moore K. E., 1995, "Simulating a UNIX Processor Using Petri Nets and Alpha/SIM", *Proceedings of the Summer Computer Simulation Conference, SCSC '95*, Ottawa, Ontario, p63~69.

Burns A. and Davies G., 1993, "Concurrent Programming", Addison-Wesley, ISBN 0-201-54417-2.

Burns R.S, 1992, "An Intelligent Integrated Ship Guidance System", *Proceedings of CAMS '92,* p199~208.

Burns R.S, 1995, "An Intelligent Automatic Guidance Systems For Surface Ships", *Proceedings of MARTRANS 95*, Plymouth, UK, p641~652.

Caillou M., 1991, "The Bridge Simulator of the Saint-Malo National Mercantile Marine School", *Navigation*(France), Vol. 39, Part 156 : p497~505 (In French).

Castelfranchi C., 1994, "Guarantees for Autonomy in Cognitive Agent Architecture", *Proceedings of the 1994 Workshop on Agent Theories, Architectures and Languages*, ECAI'94, Amsterdam, The Netherlands, August, p50~61.

Caterpillar, 1995, "Caterpillar engineers testdrive new vehicle designs in a virtual world created using CAD data and WorldToolKit", *Simulation*, Vol. 65, No 3 September, Cover Photograph.

Chen Y. H., 1992, "Ship Simulator Instruction in Taiwan.", *PhD Thesis*, University of Wales, Cardiff.

Cohen P.R., Greenberg M.L., Hart D.M. and Howe A.E., 1989. "Trial by Fire: Understanding the Design Requirements for Agents in Complex Environments", *A.I. Magazine*, Vol. 10, Part 3, p32~48.

Cohen P.R. and Howe A.E., 1988. "How Evaluation Guides AI Research", *A.I. Magazine*, Vol. 9, Part 4, p35~43.

Colley B.A., Curtis R.G. and Stockel C.T., 1983, "Manoeuvring times, Domains and Arenas", *Journal of Navigation,* Vol. 36, p324~328.

Covrigaru A. and Lindsay R., 1991, "Deterministic Autonomous Systems", *AI Magazine,* 12, p110~117.

Craig D.B., 1991, "An Object Oriented Simulation of Autonomous Agents in a Complex Physical Environment", *Proceedings of the 2nd Annual Conference on AI, Simulation and Planning in High Autonomy Systems*, April, p31~38.

Cross S.J., 1996, "Regional Progress Report South and West Europe", Presented at the 1996 International Navigation Simulator Lecturer's Conference:9, Vancouver, Canada, Aug 19th.

Curry D. M. and Combs C. A., 1995, "Requirements for Validation of STINGER Hardware-in-the-Loop Simulations", *Proceedings of the Summer Computer Simulation Conference, SCSC '95*, Ottawa, Ontario, p449~454.

Davis P.V., Dove M.J., and Stockel C.T., 1980, "A Computer Simulation of Marine Traffic Using Domains and Arenas", *Journal of Navigation,* Vol. 33, p 215~222.

Dongha P., 1994, "Commitments, Rationality, and Intention Reconsideration", *Proceedings of the 1994 Workshop on Agent Theories, Architectures and Languages*, ECAI'94, Amsterdam, The Netherlands, August, p186~199.

Durfee E. H., Lesser V. R. and Corkhill D.D., 1989, "Cooperative Distributed Problem Solving", *The Handbook of Artificial Intelligence,* Vol IV. Addison Wesley, Barr A. Cohen P and Feigenbaum (Eds).

Fenton J. and Beck K., 1989, "Playground: An Object Oriented Simulation System with Agent Rules for Children of All Ages", *OOPSLA '89 Proceedings,* October, p123~136.

Ferguson I.A, 1994, "Integrated Control and Coordinated Behaviour: A Case for Agent Models", *Proceedings of the 1994 Workshop on Agent Theories, Architectures and Languages*, ECAI'94, Amsterdam, The Netherlands, August, p186~199.

Ferranti, 1985, *Training Systems Information Pack.* Ferranti Computer Systems, Cheadle Heath Division, Stockport, Cheshire, UK. SK3 OXQ.

Findler N.V., 1991, "Distributed Artificial Intelligence Approach to Air Traffic Control", *IEE Proceedings-D*, Vol. 138, No 6, Nov, p515~524.

Fisher M. and Wooldridge M., 1993, "Specifying and Verifying Distributed Intelligent Systems", *Proceedings of the 6th Portuguese Conference on Artificial Intelligence,* Portugal.

Foley J. D. van Dam A. Feiner S. K. and Hughes J. F., 1990, "Computer Graphics: Principles and Practice",Second Edition, Addison-Wesley, ISBN 0-201-12110-7.

Fujii Y. and Shiobara R., 1971, "Traffic Capacity", *Journal of Navigation (UK),* Vol. 24, p542~552.

Gasser L. and Ishida T., 1991, "A Dynamic Organisational Architecture for Adaptive Problem Solving", *Proceedings of the 9th National Conference on Artificial Intelligence,* AAAI '91, p185~190.

Goldsmith S., 1993, "A Practical Guide To Real-Time Systems Development", Prentice-Hall, London.

Goodwin E. M., 1975, " A Statistical Study of Ship Domains", *Journal of Navigation,* Vol 28, p328~344.

Grabowski M., 1990, "Decision Support to Masters, Mates on Watch and Pilots : The Piloting Expert System", *Journal of Navigation*, Vol. 43, Part 3, p364~384.

Grabowski M. and Sanborn S., 1992, "Knowledge Representation and Reasoning in a Real-Time Operational Control System (SPES)", *Decision Sciences,* Vol. 23, No 6, November/December.

Grabowski M. and Sanborn S., 1995a, "Integration and Preliminary Shipboard Observations of an Embedded Piloting Expert System", *Marine Technology and Marine Technology News,* Vol. 32, Part 3, p216~223.

Grabowski M. and Sanborn S., 1995b, "Smart Charts - Smart Bridge Observations of the Piloting Expert System", *International Hydrographic Review*, Vol. 72, Part 1, p125~149.

Guicharrousse M., 1990, "Use of Simulators and Manned Models for Training", *The Nautical Institute on Pilotage and Ship Handling, The Nautical Institute (pubs),* 202 Lambeth Road, London.

Gurton A., 1996, "Intelligent Agents: Artifice and Artificial Life", *Computer Weekly*, October 24th. p46.

Halfhill T. R., 1996, "Agents and Avatars", *Byte*, February, p69~72.

Hanks S., Pollack M. E. and Cohen P. R., 1993, "Benchmarks, Test Beds, Controlled Experimentation, and the Design of Agent Architectures", *AI Magazine,* Vol. 14, No 4, Winter, p17~42.

Hewitt C. and Leiberman H., 1984, "Design Issues in Parallel Architectures for Artificial Intelligence", *Proceedings of the 28th IEEE Computer Society International Conference,* San Francisco, Ca., p418~423.

Hill D.R.C., 1996, "Object-Oriented Analysis and Simulation", Addison-Wesley, ISBN 0-201-87759-7.

Howe A.E., Hart D.M. and Cohen P.R., 1990, "Addressing Real-Time Constraints in the Design of Autonomous Agents", *The Journal of Real-Time Systems,* Vol. 1, p81~97.

IMO-SOLAS, 1980, "IMO Safety of Life at Sea, resolutions A.422(XI): Specification for a Marine Navigational Equipments Simulator (1980) and A.482(XII) : Amendments to the Specification for a Marine Navigational Equipments Simulator (1980) 3-11-88", *International Maritime Organisation,* London.

Ilie J-M., Maizi Y. and Poitrenaud D., 1995, "Towards an Efficient Simulation Based On Well-Formed Petri Nets Extended with Test and Inhibitor Arcs", *Proceedings of the Summer Computer Simulation Conference, SCSC '95,* Ottawa, Ontario, p63~69.

Iribarren J. R., Baonza A., Santas J. C. and Navarro A., 1995, "Moored Ship Problems in the Port of Cadiz", *Proceedings of MARTRANS 95*, Plymouth, UK, p687~694.

IRPCS, 1989, "International Regulations for Preventing Collisions at Sea, 1972 with amendments adopted from November 1989", *International Maritime Organisation,* London.

Jackson P., 1990, "Introduction to Expert Systems", Addison-Wesley, 2nd Edition, ISBN 0-201-17578-9.

James M.K., 1986, "Modelling the Decision Process in Computer Simulation of Ship Navigation", *Journal of Navigation*, Vol. 39, No 1, p32~48.

Jennings N. R., 1993, "Controlling Cooperative Problem Solving in Industrial Multi-Agent Systems Using Joint Intentions", *KEAG Technical Report No93/25.*

Kasasbeh Y., 1994, "Automatic Berthing of Ships", *Technical Report,* Presented at the Meeting of Kelvin Hughes, University of Plymouth, and John Moores University Special Interest Group in Marine Technology, November.

Kearney P. J., 1992, "Experiments in Multi-Agent System Dynamics", *Technical report, Information Technology Department, Sharp Laboratories of Europe Ltd,* Oxford.U.K.

Khedro T. and Genesereth M.R., 1994, "The Federation Architecture for Interoperable Agent-Based Concurrent Engineering Systems", *Concurrent Engineering : Research and Applications,* Vol. 2, p125~131.

Kim W. S., 1990, "Port Design and Simulation: Validity of Micro Computer Aided Port Design Simulation and its Application", *PhD Thesis*, University of Wales, Cardiff.

Kinny D. and Georgeff M.P., 1991, "Commitment and Effectiveness of Situated Agents", *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Menlo Park, California, p82~88.

Kruglinski D. J., 1995, "Inside Visual C++", Third Edition, Microsoft Press, ISBN 1-55615-891-2.

Kuroda T., Hoshino Y., Itoh H. and Seki H., 1994, "A Communication Efficiency Evaluation of Multi-Agents with Short-term Memory", *Proceedings of the 3rd IEEE International Workshop on Robot and Human Communication,* p317~322.

Lee K., Mansfield W.H. and Sheth A.P., 1993, "A Framework for Controlling Cooperative Agents", *Computer*, July, p8~16.

Leiberman H. 1996, "Letizia : An Agent That Assists Web Browsing", Media Labs, Massachusetts Institute of Technology, Cambridge, MA., URL : http://lieber.www.media.mit.edu/people/lieber/Lieberary/Letizia/Letizia.html.

Levi S. and Agrawala A. K., 1990, "Real Time System Design", McGraw-Hill International Publications, ISBN 0-07-100797-0.

Levy R. and Rosenschein J.S., 1991, "A game Theoretic Approach to Distributed Artificial Intelligence and the Pursuit Problem", *Decentralised AI 3 : Proceedings on the Third European Workshop on Modelling Autonomous Agents in a Multi-Agent World,* Demanzeau Y. and Muller J.P. (Eds), Aug, p129~146.

Lisowski, J. 1992, "Properties of Fuzzy Probability Sets in Safe Navigation", *Proceedings of CAMS '92*, p209~219.

McCabe F. G. and Clark K. L., 1994, "Agent PRocess Interaction Language", *Proceedings of the 1994 Workshop on Agent Theories, Architectures and Languages*, ECAI'94, Amsterdam, The Netherlands, August, p280~296.

McCallum I.R. 1980, "A Ship Steering Mathematical Model for All Manoeuvring Regimes", *Proceedings of the Symposium on Ship Steering Automatic Control*, p119~139.

McCleod , 1987, "The Collins Compact Dictionary", Collins, London. ISBN 0 00 433149-4

MacFadzean R., Barber K. S. and Szygenda S., 1995, "Discrete Event Simulation of Radar Defence", *Proceedings of the Summer Computer Simulation Conference, SCSC '95*, Ottawa, Ontario, p489~494.

Maitre B. and Laasri H., 1990, "Cooperating Expert Problem-Solving in Blackboard Systems: ATOME Case Study", *Decentralised AI 2 : Proceedings on the Second European Workshop on Modelling Autonomous Agents in a Multi-Agent World,* Demanzeau Y. and Muller J.P.(Eds), France, Aug, p251~263.

Maruichi T., Ichikawa M. and Tokoro M., 1990, "Modelling Autonomous Agents and Their Groups" , *Decentralised AI 2 : Proceedings on the Second European Workshop on Modelling Autonomous Agents in a Multi-Agent World,* Demanzeau Y. and Muller J.P.(Eds), France, Aug, p215~234.

Meurn R.J., 1996, "Simulated Voyages: Using Simulation Technology to Train and License Mariners", Committee on Ship-Bridge Simulation Training, Presented at the 1996 International Navigation Simulator Lecturers' Conference:9, Vancouver, Canada, Aug 19th.

Minski M. L., 1986, "Society of Mind", Simon and Schuster.

Moffat and Frijida, 1994, "An Agent Architecture : Will", *Proceedings of the 1994 Workshop on Agent Theories, Architectures and Languages,* ECAI'94, Amsterdam, The Netherlands, August, p216~225.

Montgomery T.A. and Durfee E.H., 1990, "Using MICE to study Intelligent Dynamic Coordination", *Proceedings of the 2nd International Conference on Tools for AI,* p438~444

Moon J., 1991, "Automatic RADAR Plotting Aid (ARPA) Simulation", BSc project diss., Polytechnic of Wales.

Moon J. and Tudhope D., 1995a, "MARINES a Multi-Agent testbed for marine simulation", *Proceedings of MARTRANS 95,* Plymouth, UK, p777~784. (Award for the best student paper(Awarded by Noble Denton UK))

Moon J. and Tudhope D., 1995b, "(A Multi-Agent Realm for Investigating Navigators' Educational Simulators) Introducing the MARINES Testbed", *Proceedings of the Summer Computer Simulation Conference, SCSC '95,* Ottawa, Ontario, p949~954.

Muller J.P., Pischel M. and Thiel M., 1994, "A Pragmatic Approach to Modeling Autonomous Interacting Systems -Preliminary Report-.", *Proceedings of the 1994 Workshop on Agent Theories, Architectures and Languages*, ECAI'94, Amsterdam, The Netherlands, (Aug.), p226~240

Myers B. and Doner C., 1992, "Programmer's Introduction to Windows 3.1", Sybex, ISBN 0-7821-1034-7.

Nevins M. R. Macal C. M. and Joines J., 1995, "PORTSIM : An Object Oriented Port Simulation", *Proceedings of the Summer Computer Simulation Conference, SCSC '95*, Ottawa, Ontario, p160~165.

Nielsen J., 1993, "Usability Engineering", Academic Press, ISBN 0-12-518405-0.

Norton P. and Yao P.L., 1992, "Borland C++ Programming for Windows", Bantam Books.

O'Hare G. M. P, and Jennings N. R., (Eds), 1996, "Foundations of Distributed Artificial Intelligence", Sixth Generation Computer Technology Series, Wiley Inter-Science, ISBN 0-471-00675-0

Okada T., Kuroda T. and Hoshino Y., 1993, "An Evaluation of Multi-Agent Behaviour, in a Sensing Communication World", *Proceedings of the 2nd IEEE International Workshop on Robot and Human Communication,* p302~307.

Parnas D., 1972, "On the Criteria to be Used in Decomposing Systems into Modules", Comm, ACM, Vol 15, Part 2, p1053~1058.

Pebody M., 1993, "How do you Choose Your Agents? How do you Distribute Your Processes? ", *The Biology and Technology of Autonomous Agents,* Steels L (Ed), Springer-Verlag, p345~376

Penders J.S.J.H., 1993, "Conflicts in the Spatial Interaction of Autonomous Agents", *Technical Report, NT-PU-93-1728*, PTT Research, Postbus 421, 2260, Ak. Leidschendam, Holland.

Phillips A. and Bresina J., 1991, "NASA TileWorld Manual", *Technical Report TR-FIA-91-04,* NASA Ames Research Center, Mountain View, California.

Pike R. D., 1997, "Fast Craft and the COLREGS", *Journal of Navigation,* Vol. 50, No 2, p256~260.

Pollack M.E. and Ringuette M., 1990, "Introducing the TileWorld:- Experimentally Evaluating Agent Architectures", *Proceedings of the 8th National Conference on Artificial Intelligence,* p183~189.

Polkinghorne M.N., Burns R.S. and Roberts G.N., 1992, "A Fuzzy Autopilot for Small Vessels", *Proceedings of the 2nd Conference on Modelling and Control of Marine Craft,* p349~361.

Polkinghorne M.N., Burns R.S. and Roberts G.N., 1995, "Performance Index Derivation for a Self-Organising Fuzzy Autopilot", *Proceedings of MARTRANS 95,* Plymouth, UK, p757~765.

Pourzanjani M., 1990a, "Formulation of the Force Mathematical Model of Ship Manoeuvring", *International Shipbuild. Progr.* Vol 37, No 409, p5~32.

Pourzanjani M., 1990b, "Requirements and Applications of Ship Dynamic Simulators." *Proceedings of the 1990 UKSC Conference on Computer Simulation,* Brighton UK, (Sept.), p244~249.

Pressman R. S., 1992, "Software Engineering: A Practitioner's Approach", Third Edition, McGraw-Hill International Editions, ISBN 0-07-112779-8.

Reddy M. and Moon J., 1995, "Development and Evaluation of Multi-Agent and Agent-Based Simulation Environments", *Proceedings of DIMAS '95,* Krakow, Poland, p393~401.

Redfern A. and Lin B., 1995, "Navigational Efficiency in Marine Traffic Operations in the Port of Keelung", *Proceedings of MARTRANS 95,* Plymouth, UK, p687~694

Rediffusion, 1982, "CT5 Series: Advanced Visual Simulation", Rediffusion Simulation Incorporated, 2200 Arlington Downs Road, Arlington, Texas, 76011.

Robinson M., 1991, "Through a Lens Smartly", *Byte,* May, p177~187.

Rumbaugh J., Blaha M., Premerlani W., Eddy F. and Lorensen W., 1991, “Object-Oriented Modeling and Design”, Prentice-Hall International.

Sargeant J.M., Biegel J.B. and Draman M., 1992, “Managing and Interpreting Student Actions in An Intelligent Simulation Training System”, *Computers and Industrial Engineering*, Vol.23, No1-4, p197~199.

Searle J. R., 1969, “Speech Acts: An Essay in the Philosophy Of Language”, Cambridge University Press, ISBN 0 521 09626 x.

Shneiderman B., 1987, “Designing the User Interface”, Addison-Wesley, ISBN 0-201-16505-8.

Shoham Y., 1991, "Agent 0: A Simple Agent Language and its Interpreter", *Proceedings of the 9th National Conference on AI, AAAI ‘91,* Vol. 2, p704~709.

Shoham Y., 1993, "Agent Oriented Programming." *Artificial Intelligence,* No 60, p51~92.

Smeaton G.P. and Coenen F.P., 1990, “Developing an Intelligent Marine Navigation System”, *Computing and Control Engineering Journal*, March, p95~103.

Smeaton G.P., Fawcett S.P.M., Dineley W.O.D. and Coenen F.P., 1992, “The use of Interactive Video and Spatial Data In a Marine GIS/Electronic Chart”, *Paper No 25,Presented at EURNAV ‘92 Digital Mapping and Navigation, The 9th International Conference of the Royal Institute of Navigation and the German Institute of Navigation(DGON),* London, November.

Smeaton G.P., Dineley W.O. and Tucker S.M., 1994, “ECDIS and ARPA: A Marriage of Convenience ?”, *Presented at a Meeting of the Institute of Marine Engineers*, London, 26th April.

Sommerville I., 1995, “Software Engineering”, Fifth Edition, Addison-Wesley, ISBN 0-201-42765-6.

Staniford G. and Paton R., 1994, “Simulating Animal Societies with Adaptive Communicating Agents”, *Proceedings of the 1994 Workshop on Agent Theories,* *Architectures and Languages*, ECAI'94, Amsterdam, The Netherlands, August, p135~146.

STCW, 1995, "The International Convention on Standards of Training, Certification and Watchkeeping for Seafarer's, 1978, as amended in 1995 (STCW Convention) and Seafarer's Training Certification and Watchkeeping Code (STCW Code)", International Maritime Organisation, London.

Steeb R., Cammarata S., Hayes-Roth F. A., Thorndyke P. W. and Wesson R. B., 1981, "Distributed Intelligence for Air Fleet Control", *Technical Report R-2728-ARPA.*, Rand Corporation.

Steels L.(Ed), 1993, "The Biology and Technology of Intelligent Autonomous Agents", Springer-Verlag.

Sudehendar H. and Grabowski M., 1996, "Evolution of Intelligent Shipboard Piloting Systems: A Distributed System for the St. Lawrence Seaway", *Journal of Navigation,* Vol. 49, Part 3, p362~376.

Sugawara T., 1993, "Using Action Benefits and Plan Certainties in Multiagent Problem Solving", *Proceedings of the 9th Conference on AI for Applications,* p407~413.

Thomas S. R., 1994, "The PLACA Agent Programming Language", *Proceedings of the 1994 Workshop on Agent Theories, Architectures and Languages,* ECAI'94, Amsterdam, The Netherlands, August, p307~319.

Thomson-Marconi, 1996, Sonar Simulation Information Pack, Thomson-Marconi Ltd, Cheadle Heath, Stockport, UK.

Thorne M., 1994, "Computer Organisation and Assembly Language Programming", Second Edition, The Benjamin/Cummings Publishing Company Inc., ISBN 0-8053-6879-5.

Townsend C., 1988, "Understanding C", Howard W. Sams and Company, ISBN 0-672-27278-4.

Van De Velde, W., 1993, "Cognitive Architectures-From Knowledge Level to Structurel Coupling", *The Biology and Technology of Autonomous Agents,* Steels L (Ed), Springer-Verlag, p197~221

Van Dyke Parunak H., 1996, "Applications of Distributed Artificial Intelligence in Industry", *Foundations of Distributed Artificial Intelligence,* O'Hare and Jennings(Eds), Wiley Inter-Science, 1996.

Von Martial F., 1990, "Interactions Among Autonomous Planning Agents", *Decentralised AI 2 : Proceedings on the Second European Workshop on Modelling Autonomous Agents in a Multi-Agent World,* Demanzeau Y. and Muller J.P.(Eds), France, Aug, p105~119.

Wang C. and Li J., 1996, "The development of navigation simulators and their applications", Presented at the 1996 International Navigation Simulator Lecturer's Conference:9, Vancouver, Canada, Aug 19th.

Wavish P. and Graham M., 1994, "Roles, Skills and Behaviour",*Proceedings of the 1994 Workshop on Agent Theories, Architectures and Languages*, ECAI'94, Amsterdam, The Netherlands, August, p320~333.

Wildberger A. M., 1995, "Some Implications of Soft Computing For Simulation", *Proceedings of the Summer Computer Simulation Conference, SCSC '95,* Ottawa, Ontario, p237~241.

Wooldridge M., 1994a, "Coherence in Social Action", *Proceedings of the 11th European Conference on Artificial Intelligence, ECAI '94,* Amsterdam, The Netherlands, August, p279~283.

Wooldridge M., 1994b, "This is MyWorld : The Logic of an Agent-Oriented Testbed for DAI", *Proceedings of the 1994 Workshop on Agent Theories, Architectures and Languages, ECAI'94,* Amsterdam, The Netherlands, August, p147~163.

Wooldridge M. and Jennings N.R., 1994, "Agent Theories, Architectures, and Languages: A survey", *Proceedings of the 1994 Workshop on Agent Theories, Architectures and Languages*, ECAI'94, Amsterdam, The Netherlands, August, p1~32.

Zedan H. S. M. (Ed), 1990, "Distributed Computer Systems", Butterworths, ISBN 0-408-02938-2.

Zhao J., Wang F. and Wu Z., 1992, "The Development of Ship Collision Avoidance Automation", *Journal of Navigation* (UK), Vol. 45, No 1, p107~113.

Zlotkin, G. and Rosenschein, J.S., 1989, "Negotiation and Task Sharing Among Autonomous Agents in Cooperative Domains",*11th International Joint Conference on Artificial Intelligence*, Vol.2, Detroit, USA, p912~917.

Zlotkin, G. and Rosenschein, J.S., 1991, "Incomplete information and deception in Multi-Agent Negotiation",*12th International Joint Conference on Artificial Intelligence*, Sydney, Australia, p225~231.