MODELLING TRAFFIC NAVIGATION NETWORK WITH A MULTI-AGENT PLATFORM

THIERRY HUET[1], TAHA OSMAN[2], CYRIL RAY[1]

[1] Ecole Navale – IRENav
BP 600 – Lanveoc-Poulmic, F-29240 Brest Naval, France
huet@ecole-navale.fr, ray@ecole-navale.fr
[2] Nottingham Trent University – School of Computing and Mathematics
Burton Street – Newton Building, NG1 4BU – Nottingham, England
taha.osman@ntu.ac.uk

Abstract: The increase of maritime traffic in the past few years have posed concerns about the safety of using this transport facility in the future. Recent accident statistics show that the age of boats, legislation and traffic flow control are nowadays insufficient to ensure the security of boats, crew, coasts and ecological system. Maritime authorities have set up Vessel Traffic Systems (VTS), which follows ship navigation specifically in high density areas. In this paper, we present a new computing approach to simulate and improve the security of maritime traffic. Based on agent technology and distributed systems this solution models the ship behaviours and adds to Vessel Traffic Systems the capability of comparing original and simulated ship tracks.

Keywords: Maritime Navigation Modelling, Traffic Analysis, Multi-Agent Systems, Distributed Object Computing.

1. INTRODUCTION

Sea traffic grows each years considerably. During 2001, intra-community goods transport increased by 27% [European Commission 2001]. An outcome of this growth is the congestion of different parts of the navigation's network such as the Cape Gris Nez, Ouessant in northern Atlantic and Malacca or Tsushima strait near the Asian coast. The result is a reduction of competitiveness in comparison to other means of transports and the increase of accident risks.

Disasters resulting from maritime accidents cause considerable damage and monopolize significant human and material resources [Paul 1997] in order to be solved. Relevant illustrations of the problem are the sinking of the ERIKA [BEA Mer 2000] in western coast of France which provoked a large oil leakage, and the sinking of the IEVOLI SUN [BEA Mer 2001b] which provoked chemical diseases near Chausey Islands. These accidents are mainly due to the excessive age of the ships, their bad maintenance, the crew lack of knowledge, and to non-compliance with the sea rules of transport. After the shipwreck of the ERIKA, recommendations were issued to enforce the modernization of naval equipment, improve crew training, and, more interestingly for our work, the establishment of centres for surveying the maritime traffic [BEA Mer 2001a]. In order to reduce the number of accidents, governments have two possibilities:

- The issue of regulations binding the ship's managers to increase security on board by training their staff and carrying-out regular maintenance of their ships. However, the economic market implies that if they want to survive, they have to propose the best price for freight transport. Therefore, generally, they prefer to reduce these costs.
- The survey of maritime traffic. With specific tools, national authorities can survey their territorial waters and apply national or international legislation, resulting in a cut in the amount of registered diseases.

Nowadays, many countries such as China, United States of America, Great Britain, and France have chosen a more pragmatic solution. They decided to install a large number of surveying tools and reinforce maritime legislation.

In this paper, we present a new method for modelling the maritime transport in a high traffic area. In section 2, we describe the problem of safety navigation and naval transport simulation. Section 3 presents technical problems related to the setting up of a safety navigation simulator and how they can be addressed by using new computing approaches. In section 4 we present the chosen solution we will implement to validate traffic navigation modelling. The last section presents outlooks and concludes this paper.

Proceedings 17th European Simulation Multiconference
(c) SCS Europe BVBA, 2003 ISBN 3-936150-25-7

2. RELATED WORK AND SPECIFICATION OF THE REQUIREMENTS

The majority of transportation systems are monitored by computer systems. Applications related to road network monitor vehicles, gives information about construction, accident, detours [GoodWin and Pfrang 2001] and traffic analysis [Hadouaj et al. 2000]. For Air traffic control, applications are focused on traffic survey to prevent collision [Krozel and Peters 2000]. For transportation network, information is given on scheduling [Zhu et al. 2000], coordination between railways and road network on intermodal platform [Bürckert et al. 1999].

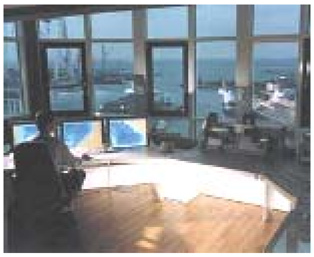

Fig. 1. Center equipped with VTS at Flint, Sweden (Öresund)

Vessel Traffic Systems (VTS, see figure 1) are used to manage maritime traffic [Amiel 2002]. It is a computer based system which uses a similar methodology to that used in air traffic control. One or several Radars are connected to the VTS and give ships bearing. Areas that are covered by VTS are bounded to ports, estuaries, straits and nearest coast. However, because of Radar and VTS costs, coastal area is not fully covered. Each ship is initially localized by Radar then, after vocal identification, the ”track” is updated.

In order to identify high density areas where an improvement of the maritime traffic security should be set up, we in a first study, establish a following of ship tracks around western Europe [Huet 2002b]. In figure 2, dots represent the true roads followed by ships and their aggregations illustrate high density ships (represented by the darkest color). One of the most dangerous area is the Channel. Indeed the huge amount of ships and the lack of complete surveys of this area (see figure 3) lead as seen recently to accident (e.g. Tricolore shipwreck). Traffic flow in this area can reach up to 500 ships a day. In Gris Nez/Dover area (eastern part of the Channel), where the width is the shortest, this high density traffic have to be supervised with care in order to prevent potential shipwrecks. In the following, we will focus on the Channel area that can be considered as the most relevant area for the improvement of traffic security using new computing solutions.

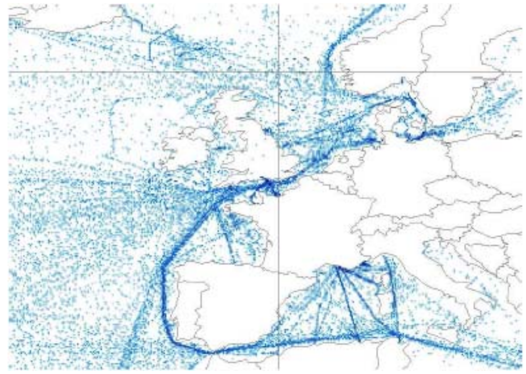

Fig. 2. Maritime traffic in Europe

In order to cover the Channel area, French and English government had set up VTS all along the coasts (see figure 3). In France, Maritime Rescue Co-ordination Center (MRCC) are fitted with VTS. On the French side, one can find a VTS associated to a CROSS (the French MRCC acronym). These are at Gris Nez, Cherbourg and Brest. In England, VTS are at Dover, Portland and Falmouth. Some VTS are connected to each other to exchange information. Hence, it is in theory possible to follow a ship from the beginning to the end of the Channel, but as illustrated in the figure 3 some areas are uncovered.

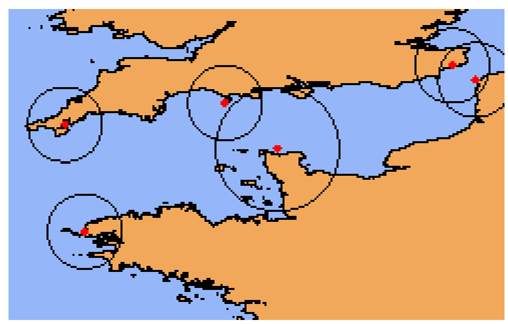

Fig. 3. Estimated area covered by VTS on the Channel

When a ship goes out of a VTS covered area, date, time, speed and direction parameters are given to the following VTS as a digital file. This file is also sent by fax to insure that the transmission has been correctly done. Then, the following VTS will propose an estimate arrival date and location. However, there is no information in-between covered areas, so if a hazardous situation arises, traditional methods of search and rescue are set up.

Adding a simulation interface in a VTS allows us to propose some additional services. In state of the art systems, VTS technology is fully automated. However, whereas following a ship on a screen is controlled, simulating its movement is yet a research topic. Therefore, For VTS systems it could be useful to: (1) estimate the vessel road when outside the radar cells; (2) compare the ”true road” to the ”estimated road” to anticipate potential problems; (3) replay a true situation for training; (4) simulate a ”day traffic” to estimate speed limitation, maximum density.

Optimum vessel track depends on different parameters such as weather conditions, human interaction or vessel geometry/attitude, and vessels interaction between themselves. In this context, we propose to model each ”track” as a computer agent whose location is the major parameter.

3. ENABLING SAFETY NAVIGATION SIMULATION

Setting up such a safety navigation simulator requires taking into account lots of parameters. Indeed, to model the vessel movement in a distributed context, we have to take into account algorithms complexity. [Huet 2002a] demonstrates an agent technology that give the same algorithm complexity but increase interaction between user and model. We also have to take into account computer performances and operating systems [Ray and Huet 2000] and specifically the scalability problem. To avoid loosing information, fault tolerance procedures should be used.

The agents computing paradigm fulfils most of the afore mentioned requirements. Agents are autonomous software entities that roam the Internet to perform tasks on behalf of the user. They access network resources more efficiently because they can move to the resources location rather than exchanging multiple network messages over congested bandwidth. Agent systems are also ideal for modelling interrelated objects behaviour. They use speech-act based communication primitives to cooperate [O’Brian and Nicol 1998], while maintaining their autonomy. The mobility and autonomy of mobile agents allow them to trail the track of ships, possibly migrating to other computing nodes that are closer to new ship tracking locations, while the advanced agent communication primitives should allow for the seamless integration of safety navigation rules into the cooperative behaviour of the agents representing the ships.

Despite their promise, there are a lot of issues such as trust, scalability, reliability that need to be resolved before agent systems can be adopted by distributed processing and Internet-based applications. In this work, we focus on problems that have direct implications on the safety navigation of marine traffic. The choice of an agent-based support for modelling of the various entities interacting in a maritime environment has some drawbacks. From an implementation point of view, the autonomy of each agent is ensured by the creation of an individual process or a thread.

Unfortunately computers have limited resources resulting in a limited number of manageable processes (threads) and consequently agents. An advantageous configuration of a safety navigation simulator should imply the modelling of one ship using one agent because agent properties and behaviours can naturally replicate ship behaviour. This modelling approach is judicious but raises the problem of scalability of such a simulated system.

As mentioned earlier, computing systems have limited resources, thus a limited number of ship that can be modelled. In a realistic simulation of ship navigation in the Channel one can expect about 500 boats to manage by day. This figure is not so far from common limits of agent number that can executed correctly by a computer (this limit depends on the computer itself but also on the operating system) and can lead to an overhead on agent computing engine. This overload is reinforced when the navigation space is larger or when the traffic increases. In order to provide an efficient simulator, this scalability problem needs to be addressed in order to ensure that any solution doesn’t fall behind real-world response time.

Recent advances in computing offer new solution for the simulation of large-scale systems such traffic navigation system. The use of distributed system for efficient computing support for multi-agent platform allows to overcome the scalability problems. This distributed approach has been already used for different applications. In the domain of the understanding of urban phenomenon, a distributed agent approach is used for the simulation of traffic [Erol et al. 1999]. In [Ray and Claramunt 2002] a distributed platform models and simulates disaggregated data flows in an airport terminal.

The second major challenge is safeguarding the agent computing engine. Marine navigation systems are safety critical and faults affecting the driving application can have disastrous consequences. We mentioned earlier that agents are autonomous software objects that can migrate between Internet hosts and communicate with each other via sophisticated message exchange primitives to achieve a common goal, for example collision evasion. Hence, to ensure the reliability of the navigation system, we need to make the computing engine tolerant to faults that might affect the agent execution or the communication channel between the agents [Osman and Bargiela 2000a].

4. SOLVING SAFETY NAVIGATION

4.1 Ship modelling

The ship, identified by its location is the element, which is moving according to many dynamically changing environmental factors. Here we propose to model each ship by an agent. Since agents have builtin support for multi-threaded behaviour, they can integrate sea/weather information (which is environmental information), vessel geometry and human interactions (figure 4).

We can find, in all navigation books, information to calculate the ship road while taking into account wind or stream drift. We will use this information to model the ship behaviour. This information depends on the type of navigation. Coastal navigation follows code of nautical procedures and practices, lights, buoys and fog signals, separation traffic scheme and sailing directions while ocean navigation follows shortest way rules (to go from one point to another one, the fastest is the shortest) and sailing directions.

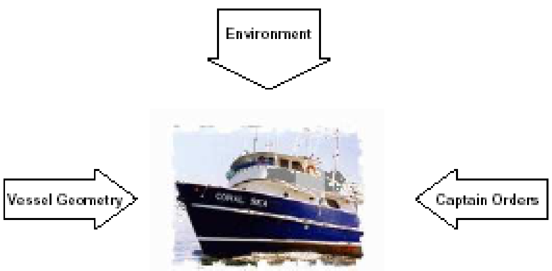

Fig. 4. Modelling interactions between the ship and its environment

In [BEA Mer 2001b], events which caused the loss of the IEVOLI SUN are described. Major remarks in the report were the sea and weather condition, state of the ship, choice done by the captain to find the best road and ship condition:

1) Human interaction: the captain has to take into account several parameters to define its road. For instance, he has to take into account the distance between overlapping ships, chart information (such as traffic separation device).
2) Vessel Geometry/attitude: each vessel has its own performance parameters such as manoeuvrability, gyration, distance to stop or wind catching, stream catching, draught.
3) Environmental information integrates relationships between the ship behaviour and natural elements such as wind or tide effect.

We can identify currently several rules, which will be used in our modelling. These rules will be changed and expanded in the second step of our work. In a simplified case study, we will start by the modelling of ocean navigation. In that way, we can write:

$$\exists \lambda = \{\phi, G\}; \phi \in \{-180..180\}, G \in \{-90..90\} \quad (1)$$

λ is the ship location identified by its longitude (G) and its latitude (Φ)

$$\Lambda = \{\lambda_1 \cdots \lambda_n\};\ n \in \aleph^+ \quad (2)$$

Λ is the surveyed area.

$$\exists R = \{WP_1 \cdots WP_n\} / WP_i \subset \Lambda;\ i \le n \quad (3)$$

A road is defined by an ordered list of waypoints.

$$\exists \alpha \in \{-90 \cdots 90\},\ S \subset \Lambda;\ WPi \in R \quad (4)$$

$$\Rightarrow \tan \alpha = \frac{G_i - G}{\phi_i - \phi} \quad (5)$$

$$\Rightarrow \delta = \sqrt{(G_i - G)^2 + (\phi_i - \phi)^2};\ \delta \in \Re^+ \quad (6)$$

A ship road is defined by an azimuth **α** and distance **δ**

$$\exists \mathrm{O} / \mathrm{O} \subset \Lambda \ and\ \forall S \in \Lambda; S \notin \mathrm{O} \quad (7)$$

An obstruction area is an area in the surveyed area where ships cannot advance.

$$\exists S_p \Rightarrow S_P \subset \Lambda = \{S, f(v)\} \quad (8)$$

Each ship has its own protection area. It is a circle whose the diameter varies with the speed **v**. A protection area is an obstruction. The protection area must not intersect with another obstruction. To ensure the correctness of the working model, we propose to integrate the model components step by step.

4.2 Space modelling

Because of the geographical position of different entities (e.g. VTS) over a maritime space, we need to partition the navigation space into smaller areas in order to replicate real world. This is also needed, as mentioned to overcome the scalability problem. This maritime environment can be modelled using a hierarchic approach [Bargiela 2000], which organizes information at different level of granularity. The hierarchic decomposition is particularly adapted for the modelling of real world system that have to be implemented on distributed system. This description of the different elements constituting the environment shows three essentials components. First, this scheme allows for the identification of geographical entities of the model (nodes of the tree, figure 5). Secondly, this scheme shows dynamic behaviour of these entities. Finally it shows the entity that can be mapped on the computer for the distribution (a partition).

Partitions are managed by space managers that coordinate start up and inter agent communication. They also facilitate agent migration as modelled ships move into their neighbouring partitions. Let us notice that a *buffer zone* is needed if space manager has limited view of the space it manages (existence of uncovered area).

There are several benefits of such a distributed approach: (1) it provides manageable computing object entities (agents) in order to solve scalability.

(2) it provides local reference for agent communication and thus more efficiency in communication because objects in the same area are on the same computer. (3) it reflects real structure of how sea space is managed.

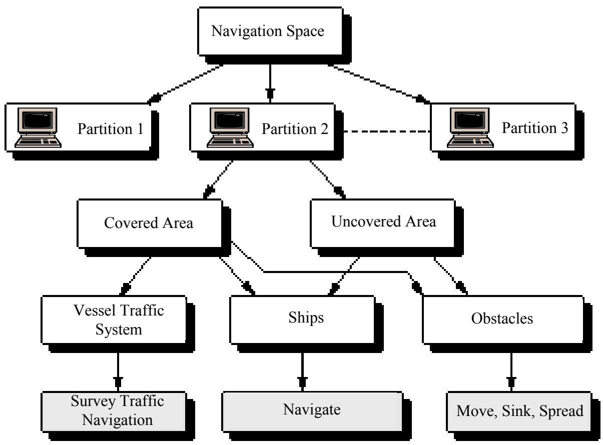

Fig. 5. Space modelling

4.3 Reliability concerns

We mentioned that naval navigation systems are safety critical, which poses reliability concerns on the proposed agent-based computing engine. The mobile agent technology is a software revolution, its goal is to utilize networked resources in a more opportunistic fashion than the traditional client/server systems, but the hardware platform on which the agents run doesn't differ from that exploited by the traditional distributed computing paradigms. Hence, the reliability of agent-based systems is affected by faults to the external agent environment, i.e. the networked hosts and the communication channel. Transient failures, caused by a temporary memory fault for example, might affect the executing agent only, while permanent faults caused by host failure will crash the running agent platform and all the executing agents. The integrity of inter-agent messaging and can be violated by faults in the communication link as well as the failure of the sending or receiving agent. Host and communication link failures can also disrupt agent migration. Our navigation model presumes that the ship agents must exchange information for track correction, collision evasion, etc., therefore we also need to address the complex problem of coordinating the recovery of the mobile agents in a collaborative environment. Here we must take into consideration how the failure of a single agent or a communication transaction can affect the consistency of the global state of collaborative agents applications. Classic distributed fault-tolerance issues such as domino effect and duplicated messages [Osman and Bargiela 2000b] are also relevant for such agent applications.

There are two main approaches to fault-tolerance of distributed systems: replication and checkpointing. Replication techniques rely on executing replicas of the application processes (agents) on redundant hardware, then the application should be able to continue executing reliably as long as at least one replica is alive. In-principle, replication techniques should have smaller overhead than checkpointing methods, that can block the application execution while retrieving a previously stored state of the agent (checkpoint) from stable storage. However, we argue that agent computing is not suitable for high-performance applications with strictly constrained response time for which such overhead can be regarded intolerable. The agent code has to be interpreted to support the portability necessary for agent mobility, which slows-down the performance of the agent code in comparison to executables that are pre-compiled into native machine code. There is also the overhead of marshalling/unmarshalling the execution code (byte-code serialization) and agent load/start-up time as agents travel their itinerary to fulfil a user task [Schill et al. 1998]. The advantage of using the agent-computing paradigm for naval navigation paradigm is in the multi-agent systems ability to logically map into the dynamic topology of roving objects (ships) and intelligent processing of their interaction in the maritime environment, rather than delivering exceptionally high performance.

Checkpointing offers a low-cost alternative to agent systems reliability, where live replicas running on redundant hardware are not required [Silva et al. 2000]. Checkpointing is easer to implement, as the management of consensus between many replicas is not required. It also fits naturally into the agent-computing model since serializing the agent code in preparation for migration effectively constitutes taking a checkpoint.

There are many flavours of checkpointing and rollback recovery: messaging logging, consistent checkpointing, and hybrid methods. The choice and detailed design of the rollback recovery protocol will fundamentally depend on the study of execution, communication, and reliability requirements of the navigation system navigation system, which will prevail in the next stage following this feasibility study. Many implementation decisions for the fault-tolerance layer will also largely depend on our choice of mobile-agent platform. The requirement to support mobility, agent tracking, message forwarding, etc. has made the agent platforms very complex middleware that significantly varies from one implementation to another. For example some agent platforms rely on weak migration [Rothermel and Schwehm 1999], i.e. proxies are used at the remote platform instead of physically de-serializing and transporting the agent code. Weak migration significantly complicates the fault-tolerance protocol since the agents no longer have autonomous execution state.

5. CONCLUSIONS

In this position paper we presented simulation for safety navigation and identified technical limitations we encountered during the first step of an implementation of our model.

In the first part of this document, we explain why it is useful to simulate traffic navigation and presented the problem of sea traffic congestion and why we need to address it. In the second part, we criticize the traditional methods currently used to survey the ships and suggest how it can be improved by applying the multi-agent technology to represent the simulated vessels. We identify probable technical problems facing the agent-based simulator such as algorithm complexity, reliability, resource management and scalability. Then, we described a modelling approach to maritime navigation that integrates ships behaviour, and suggested a multi-agent distributed systems approach for implementing the proposed model. We explained how the autonomous and mobile nature of the multi-agent computing paradigm perfectly fits into space modelling of maritime navigation.

The next stage of this project will involve implementing the navigation simulation model on a multi-agent computing bed and testing a collision alarm system based on the proposed model. We will also perform an evaluation study of the scalability and availability results of the simulation by contrasting them to real-life data.

ACKNOWLEDGMENT

We would like to thanks the French Ministry of Foreign Affairs (Alliance Program) and the British Council for their financial support.

REFERENCES

AMIEL, L. 2002. Opening up VTS systems. In *New Computer Tools for Safety Navigation – First French/Chinese Workshop*. French Naval Academy and Shanghai Maritime University, Shanghai, Shanghai Maritime University, China.

BARGIELA, A. 2000. Strategic directions in simulation and modelling. Tech. rep., The Nottingham Trent University, United Kingdom. january. UK Computing Research Strategy Meeting.

BEA MER. 2000. Rapport d'enquête sur le naufrage de l'ERIKA. Tech. rep., BEA Mer/Ministère des Transports. october. 14 pages.

BEA MER. 2001a. Rapport annuel. Tech. rep., BEA Mer/Ministère des Transports. june. 4 pages.

BEA MER. 2001b. Rapport d'enquête sur le naufrage de l'IEVOLI SUN. Tech. rep., BEA Mer/Minist`ere des Transports. december.

BüRCKERT, H.-J., FUNK, P., AND VIERKE, G. 1999. An intercompany dispatch support system for intermodal transport chains. Tech. Rep. TM-99-02, Deutsches Forschungszentrum f¨ur K¨unstliche Intelligenz. 12 pages.

EROL, K., LEVY, R., AND WENTWORTH, J. 1999. Application of agent technology to traffic simulation. Tech. rep., U.S. Department of Transportation, McLean, Virginia, USA.

EUROPEAN COMMISSION. 2001. White Paper, European Transport Policy for 2010. Tech. rep., Directorate- General for Energy and Transport. september. 22 pages.

GOODWIN, K. AND PFRANG,M. 2001. Web delivery of real-time GIS-based traveler safety information. In *14th Annual Geo-Spatial Information Systems for Transportation Symposium*. Bureau of Transportation Statistics, Crystal Gateway Marriot, Arlington, Virginia, USA.

HADOUAJ, S. E., DROUGOUL, A., AND ESPI ´E, S. 2000. How to combine reactivity and anticipation : the case of conflict resolution in a simulated road traffic. In *MABS'2000 Workshop*, A. Bundy, Ed. LNAI. Springer Verlag, 82–xx.

HUET, T. 2002a. Am´elioration des performances : ´ Etude de la complexit´e de la s´election des sondes. Tech. rep., IRENav. june.

HUET, T. 2002b. Using internet to survey the maritime traffic. Tech. rep., Ecole navale – IRENav. december.

KROZEL, J. AND PETERS, M. 2000. Decentralized control techniques for distributed air/ground traffic separation. Tech. rep., Nasa Ames Research Center. june.

O'BRIAN, P. AND NICOL, R. 1998. Fipa – towards a standard for software agents. *BT Technology Journal 16,* 3.

OSMAN, T. AND BARGIELA, A. 2000a. Fadi: A fault-tolerant environment for open distributed computing. *IEEE Proc. on Software 147,* 3, 91–99.

OSMAN, T. AND BARGIELA, A. 2000b. Fault-tolerance for mobile agent systems and applications. In *Proc. Of 2000 Workshop on Agent-based Simulation*. 211–221.

PAUL, L. 1997. A vessel traffic system analysis for the korea/tsushima strait. In *Energy-Related Marine Issues In the Regional Seas of Northern Asia*. Berkeley, California, USA.

RAY, C. AND CLARAMUNT, C. 2002. A new distributed computing approach for the modelling and simulation disaggregated data flows: application to an airport system. In *Geoide International Colloqium on Integrated Land-use and Transportation Models*. Québec, Canada, 14 pages.

RAY, C. AND HUET, T. 2000. Active objects in a distributed environment. In *Middleware 2000*. New York Palissade, U.S.A. poster session.

ROTHERMEL, K. AND SCHWEHM, M. 1999. Mobile agents. In *Encyclopedia for Computer Science and Technology*, M. D. Inc., Ed. Supplement 25, vol. 40. New York, 155–176.

SCHILL, A., HELD, A., B ¨OHMAK, W., SPRINGER, T., AND ZIEGERT, T. 1998. An agent based application for personalised vehicular traffic management. In *Proc. of the 2nd International Workshop on Mobile Agents*, K. Rothermel and F. Hohl, Eds. Lecture Notes in Computer Science. Springer-Verlag, Berlin, Germany, 99–111.

SILVA, L., BATISTA, V., AND SILVA, J. 2000. Fault-tolerant execution of mobile agents. In *Proc. of the International Conference on Dependable Systems and Networks*. New York, USA, 135–143.

ZHU, K., LUDEMA,M., AND HEIJDEN, R. D. 2000. Air cargo transport by multi-agent based planning. In *Proc. of the 33rd Hawaii International Conference on system Science*.

Dr Thierry Huet is Lecturer in Computer Science at the French Naval Academy. He is currently working at the Research institute of the French Naval Academy (IRENav) on safety navigation applications. He has been responsible for the succes of several GIS and software development project for European (CORINE, MEDSTAT projects), National (CNES, the french space agency) and regional (SIGMIP project) governmental organisations. Among its interest are co-operative and distributed network for solving problems, reuse of software components, time and GIS.

Dr Taha Osman is a lecturer in the School of Computing and Mathematics at the Nottingham Tent University. H received a B.Sc honours degree in Computing from Donetsk Polytechnical Institute, Ukraine in 1992. He joined the Nottingham Trent University in 1993 where he received an MSc in Real-time Systems in 1994 and was awarded a PhD in 1998 for his work on developing fault-tolerant distributed processing environments. His current research interest is focused on the utilisation of mobile agent systems for developing robust, open-access wireless computing environments.

Cyril Ray received a M.Sc. in computer science from University of Rennes in 1999. He is currently a PhD student in computer science at the French Naval Academy Research Institute (IRENav).

From 1998 to 1999, he worked as a student at IRISA (INRIA Rennes) where, his research activities focused on software engineering and distributed systems. Within the COMPOSE team he took part to the development of profiling tools for code specialisation and automatic generation of optimised Java Bytecode. Then, within the ADP group he worked on taxonomies of distributed shared memory, and on a new protocol which aims to describe a fail-safe distributed shared memory. Within the GIS group at French Naval Academy, his Ph.D. research is oriented towards distributed simulation of complex system and the development and application of qualitative discrete-event simulation applied to the modelling and simulation of transportation systems.