Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

YACHTS—YET ANOTHER COOPERATIVE HIGH LEVEL ARCHITECTURE TRAINING SOFTWARE

Agostino G. Bruzzone

Via Opera Pia, 15
Department of Production Engineering
DIP – University of Genoa Italy
Genova, GE 16145 ITALY

Roberto Mosca

Via Opera Pia, 15
Department of Production Engineering
MISS – University of Genoa Italy
Genova, GE 16145 ITALY

Roberto Revetria

Via Cadorna, 2
C/o Logistic Lab - Savona Campus
Liophant Simulation Club
Savona, SV 17100 ITALY

ABSTRACT

The paper proposes a new tool for supporting educational and professional skill development in HLA environment; the application proposed by the authors is devoted to provide a realistic case and an easy to understand/modify example where to extend technical knowledge of HLA.

1 INTRODUCTION

HLA is a interesting new area for simulation applications, however to learn the technical details related with this architecture and to get the skills for developing complex applications is still an interesting challenge. This is mostly due to the fact that today there is a lack in available technical tutorials; so the authors decided to implement a complete new educational tool that allows students to improve their skill in design and managing HLA Federation.

The educational support (see Figure 1) provides an application case for teaching the methodology in reference to a realistic application.

The application simulates sailing procedure of a military naval fleet as well as the weapon procedure.

The naval example is a very useful application field for HLA, in fact every ship can be modeled just thinking of it as an object with its position, velocity and direction, the torpedo can inherit for the ship object its attribute plus set of warhead characteristics.

In the next future the authors expect to include in the model submarine objects as well as airplane models.

Every federate was implemented in Java language in order to achieve some advantages in the graphical interface, but the Run Time Infrastructure was kept as C++ implementation.

Each federate was allowed to direct its ship and see the other federate's ship. A set of C++ implementation of the same model were used for simulate the background freight traffic, CGF (Computer Generated Forces) and to test interoperability between different models.

The experimental results showing great improvement in both understanding and programming complex HLA model as well as a good introduction for people that were on the first time introduced to this new simulation protocol.

2 NAVY COMBAT PROCEDURE SIMULATION

Military ships are used as protection for Cargo and Tankers during sailing hazardous waters. In such contest Navy must protect the freight ship and prepare to fight every aero-naval force that can sink or destroy ships.

The maritime weapons include: Rockets, Guns, Torpedoes and Heavy Machine Guns.

Every kind of weapon is an object with a warhead's position, velocity, direction and a destroying range.

Damages between ships and weapons may occur only if ships' dimensions and weapons' destroying range have a non empty intersection during all the period in which every warhead is active.

After this period the warhead is inoffensive and can resign from federation.

Collision, damages, losses and casualties are reordered in every model according to the severity of the combat event and broadcasted to the other federates by means of interactions.

The inheritance concept in HLA and late in OOP is the key of all implementation, in fact ships and warheads have similar behavior: every object must be able to determine its position according its motion law and be informed of the other's position according to its sensor system.

Position is a general published and subscripted attribute for each object, every object has an internal list of the other discovered objects that are updated in their attributes through RTI services.

Objects are responsible for collision detection and for graphical representation of the battlefield; in the front end of the software is possible to see all the discovered object in different colors according to their nature.

A set of pre defined operation can be accomplished, for example is possible to direct a torpedo attack or to engage a combat.

Time Advance policy was chosen as Time Stepped with an interval calculated as shown in the next formula. The basic idea was to find a ΔT in which a single ship cannot change its position much than its dimension. In such way is possible guarantee that a torpedo or a warhead fired it will hit the target during that ΔT.

$$\mathrm{Max}\ (\omega_i^{max})\Delta T < \min \lambda$$

Where ΔT is the time step duration, ω_i^{max} is the maximum speed allowed for the HLA object i and λ is the minimum HLA object i dimension.

In such way in the defined ΔT no HLA object will move of a quantity greater than its minimum dimension so any weapon will not miss its target in the middle of a simulation step.

Background freight traffic may suffer losses and casualties that are reported at the end of the simulation runs and are used to collect statistics.

3 YACHTS FEDERATION OBJECT MODEL

The entire federation has been modeled by introducing a generic object Vessel and an interaction called Message.

The HLA object Vessel has basically 6 attributes:

- Name
- Position X
- Position Y
- Position Z
- Direction
- Speed

Every Vessel is responsible for updating its position, direction and speed; the name is unique and well defined at the beginning of the simulation.

The RTI implementation was chosen as RTI1.3v7 from developed by DMSO; the single federates were implemented in Java by using the RTI Java Binding 1.3r4567.

Java 1.1.7 offer a user friendly environment useful to build simple but effective graphical interface (see Figure 2) and guarantee a structured approach to the all problem.

In fact due to the complexity of the HLA programming is much simple to understand Java code than a usual C++ implementation, as the YACHTS project is devoted to improve understanding and promote professional skill in HLA the authors decide for a complete Java front end implementation.

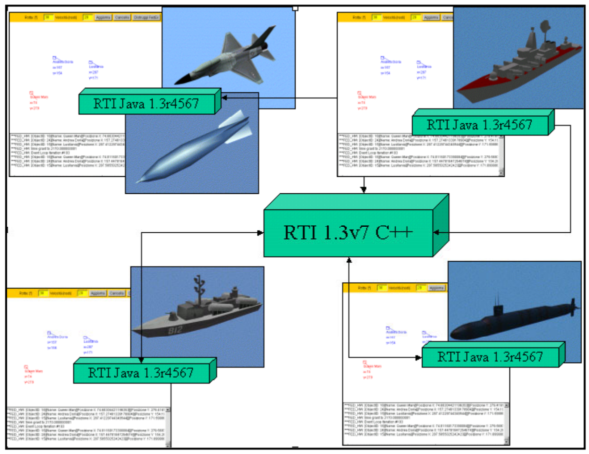

Figure 1: Implemented Federation YACHTS

Inheriting from Java object Vessel YACHTS introduce other Java object:

- Battle Ship,
- Submarine
- Torpedo
- Battle Plane
- Deep Charges
- Gun Projectiles

These Java object are the major entities in the federation and each of them make a reference to the HLA object Vessel, as in Figure 1.

Since a Battle Ship has the general Vessel behavior: it can move, live and combat in a 2D environment. Submarines and Battle Planes have the ability to move in a 3D environment and in such way they offer a more complex behavior; Torpedoes are a particular kind of Java object Vessel that has a fixed direction and a higher speed (normally a Battle Ship sails at 30 knots and a Torpedo at 45 knots) as well as a activation time.

Deep charges are allowed to move only in the Z axis (first approximation) and have their own activation rules.

Figure 2: Implemented Federation YACHTS

Gun projectiles has a parabolic motion from the firing ship to the target.

Battle Ships may launch an attack just firing torpedoes against other Ships and shooting projectiles to planes.

Submarine are allowed to attack Ships firing torpedoes and have no defense against Bombs except escape from their active range.

In case of collision, explosion an interaction is broadcast informing each federate that something is appended to the battle field. If damage suffered from an object exceed a fixed limit the object is removed from the federation.

4 YACHTS JAVA FRONT END

The essential front end is divided in 3 main panels:

- Action Panel
- Battle Field Representation
- Message Box

The last panel is used both by RTIAmbassador and FedeateAmbassador to describe the federation activities such as joining, resigning and update attributes.

The RTI callback are visualized in this area as well as the interaction message and the value of the Federates' Attributes.

In such way the operator is informed also of the background events and can better understand the HLA transactions.

Battle Field Representation, as in Figure 2, is designed to look alike a tactical table with a simplified and symbolic representation of the various objects. Colors are used to identify the nature of the objects as well as shape is used to determine the nature of the weapons used.

Drawing procedure is part of the Java object Vessel implementation and a dynamic list of the discovered federation objects is kept by each federate in order to create its own battle field scenario.

At the beginning of the simulation the first federate start the fedex process on its machine, ask for federation creation and try to join the federation itself.

Other federates now discovered that the federation execution already exist and try to join it.

Each federate create its HLA object Vessel and register as well as publish and subscribe all attribute, after successfully achieved this point it start to update its attribute values.

As soon as a new HLA object Vessel instance (ship, subs, torpedoes, deep charges, warhead or plane) is discovered, subscribing federate start to create a new Java object Vessel with empty attributes awaiting the first updating in order to complete the knowledge about the new entity.

Graphical representation of the battle field simply repaint every simulated step the federate Java object vessel itself and all the discovered instances in the discovered Java object Vessel List.

Implemented federation time management is strictly conservative (each federate is both Time Constrained and Time Regulating) and the advance time request is time stepped , future implementation will require a event based simulation in order to save simulation time.

5 APPLICATION OF YACHTS

The first operative demonstration involved 10 federates over a local area network and was successfully tested in HLA labs over Windows NT platforms.

The implemented federation was succesfully used as support for the HLA Base Training Course at University of Genoa Department of Production Engineering. During this experience the tool has demonstrate great skill in helping people both from academia and industry to understand the basic principle of HLA as well as giving a good attitude in implementing small simulation federation.

The training team, 43 people in total, was composed from 8 different institution coming for 30% from industry and the rest from other university.

People presented different experience in developing HLA federation, as in Figure 3, in effect: 15% of the participants showed some skill in programming HLA, another 65% had only theoretically knowledge about the general HLA approach, 15% (the most part from industry) had no idea about what HLA was. Only 5% of total participants had only some knowledge in the principle of HLA but without any practical experience.

The HLA Base Training Course took place in just two full day based on a main theoretical lesson plus a full day practical experience on YACHTS and determine that the 85% of the total participants receive good results in the final examination that was a simple Java federation implementation.

Such implementation was carried out using the DMSO TestFederate tool in which every participant was asked to build a Ship federate able to interact with the YACHTS module. The TestFederate is a educational tool used to perform test on federation without hard-coding any line of software but just call services on the RTI simply providing arguments and parameters in forms.

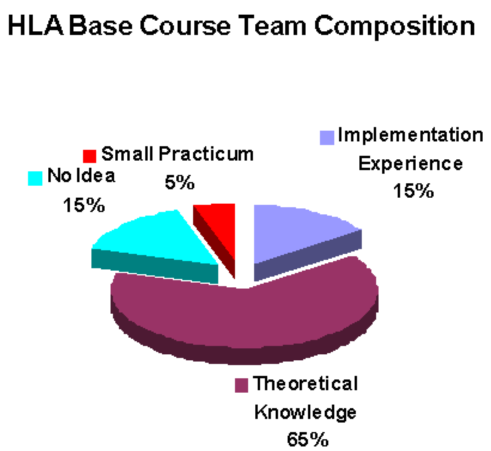

Figure 3: HLA Base Course Team Composition

The purpose of such activity is to assure that every participants had properly understood the structure of a HLA federation as well as its life cycle and provide some skill in understand which services are necessary in order to build a HLA federate.

Join and resign procedures are also explained and used in YACHTS tutorial sessions since the tool itself provide a way to create, join, resign and destroy the federation.

6 CONCLUSIONS

The tool proposed is a first step in the direction to create a real technical support for people that requires to work in this sector; the experience with the class rooms tested demonstrates that YACHTS is very effective in improving the real HLA know-how.

Authors are now planning to extend the YACHTS in order to integrate the notion of Region and the Discrete Event Time Advance; so YACHTS has proven to be very effective to support HLA Base training course as well as grant a complete list of learned lessons from HLA tutorial.

REFERENCES

Banks J., Carson J., Nelson B.1996. *Discrete-Event System Simulation*, Second Edition, Prentice Hall

Bruzzone A.G., Giribone P., Revetria R. 1999. Design and management issues in the development of the new workshops using web-based simulation and neural networks, *Proceedings of FAIM'99*, 265-276 Tilburg NE., June 23-25.

Bruzzone A.G., Giribone P. 1998. Decision-Support Systems and Simulation for Logistics: Moving Forward for a Distributed, Real-Time, Interactive Simulation Environment, *Proceedings of the Annual Simulation Symposium IEEE*, Boston, 4-9 April.

Dewar J.A., Bankes S.C., Hodges J.S: *Credible Uses of the Distributed Interactive Simulation Dis. System*, the Lucas Rand Corporation; ISBN: 0833023039.

Geraghty R., Joyce S., Moriarty S., Noone G., Joyce S., *COM-CORBA Interoperability*, Prentice Hall ISBN 0130962775.

Hamilton John A., Jr, Nash David A.,. Pooch Udo W. *Distributed Simulation*, CRC Press ISBN: 0849325900 MISS- UCChico - HLA Library.

Kenneth C. Laudon, New York University: *Management Information System Organization and Technology in the Networked Enterprise* ,sixth edition.

Kuhl F., Weatherly R, Dahmann J., Kuhl F., Jones Anita, HLA: *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*, Prentice Hall; ISBN: 0130225118.

Lorenz M. 1993. *Object-oriented software development: a practical guide*, Prentice Hall Object-oriented series.

McConnell J. 1996. Managing Client/Server Environments: tools and strategies for Building Solutions, Prentice Hall.

Paterson, D. J., Anschuetz, E., Biddle, M., Kotick, D., Nguyen, T. 1997. Simulation Interoperability Standard Organization Extension of RTI to Construct a Simulator over DistributedPC's, Proc.Spring Simulation Interoperability Workshop.

AUTHOR BIOGRAPHIES

AGOSTINO G. BRUZZONE began his engineering studies at the Italian Naval Academy with the Faculty of Pisa in 1984. After successfully completing this phase, he transferred to the University of Genoa where he earned his degree in Mechanical Engineering.Since 1991, he has taught "Theories and Techniques of Automatic Control" at the Technical High School of Finale Ligure and has become a member of the industrial simulation work group of Prof. Mosca and Prof. Giribone at the University of Genoa.He has utilized these techniques in the harbor terminal, maritime trade and sailboat racing sectors. He is currently working on a research project involving new modeling designs, AI techniques and DOE (design of Experiments); particular attention is being focused on the application of Neural Networks and Fuzzy Logic to industrial plant problems using Simulations and Chaos Theory. Currently he teaches "Project Management" and "Industrial Plants" in DIP (Mechanical and Management Engineering) Genoa University.

ROBERTO MOSCA is Full Professor of "Industrial Plants Management" and "Economy and Business Organization" at the Institute of Technology and Mechanical Plants, University of Genoa. He has worked in the simulation sector since 1969 using discrete and stochastic industrial simulators for off_line and on_ line applications. His research work focuses on the evaluation of simulation languages and new modeling techniques and his research team is developing new AI applications for industrial plant management. Currently he is directly involved as coordinator in the construction of a new campus in Savona, focused on industrial engineering.

ROBERTO REVETRIA earned his degree in mechanical engineering at the University of Genoa and he completed his master thesis in Genoa Mass Transportation Company developing an automatic system integrating ANN (Artificial Neural Networks) and simulation with the ERP (Enterprise Resource Planning) for supporting purchasing activities.He had consulting experience in modeling applied to environmental management for the new Bosch plant facility TDI Common Rail Technology in construction near Bari. During his service in the Navy as officer, he was involved in the development of WSS&S (Weapon System Simulation & Service) Project. He is currently involved, as PhD student and reseacher in the DIP of Genoa University, working on advanced modeling projects applied to ERP integration and maintenance planning applied to industrial case studies (Contracting & Engineering and Retail companies). He is member of ANIMP, Rotaract, SCS and Liophant Simulation Club.