AN HLA FEDERATION FOR EVALUATING MULTI-DROP STRATEGIES IN LOGISTICS

Roberto Revetria
MISS Genoa – University of Genoa
Via Opera Pia, 15
16145 Genoa Ge, Italy
e-mail: revetria@itim.unige.it

P.E.J.N. Blomjous, S.P.A. van Houten
Faculty of Technology, Policy and Management – Delft University of Technology
Jaffalaan 5, 2528 BX
Delft, The Netherlands
e-mail: s.a.vanhouten@student.tbm.tudelft.nl

Abstract

Distributed Simulation has proven to be very effective in modeling complex system. Since High Level Architecture has turned to a strict DoD requirement several civil domain application have been experienced. Despite the high potential of the methodology the Industrial Community is facing some resistance in implementing HLA due to the lack of directly applicable tool. In this paper authors proposes a federation for supporting Supply Chain Management by integrating a set of Arena™ based simulator able to support the optimization of a Multi Drop Delivery problem using the Capacitated Routing Vehicle Problem (CRVP). The paper outline the main issues of the Federation Design and Implementation as well as a Real Life Application of the proposed methodology.

Introduction

The recent developments in Manufacturing have boost up the practice of outsourcing in which suppliers are continuously specializing and improving in order to meet the market changes. This point is not only a way to reduce production costs but is also a common practice to increase the flexibility to the market: new ideas and proof of concept can be obtained from the "external world" and transformed into real product. The "externalize & specialize" approach has turned in to several Spin Off Projects in which Companies can gain business performance from previously critical division. A production line that is underused can be turned into an interesting business unit by transforming it into a separate company and open to the open market (i.e. PUMA Experience for Gas Turbine Power Plants Maintenance). This issue requires a efficient level of control that is becoming extremely complex for highly distributed Manufacturing Systems, in which Simulation is largely used due to the extremely non linear nature of the problems. Simulation, here, is often used to improve process. Until now all the members of a supply chain had to simulate separately their processes and information are not shared among the Supply Chain Partners But for taking advantage from the proposed methodology a full scale simulator model may be used in order to to build a model that resembles reality more effectively. The High Level Infrastructure (HLA) is a standard framework that supports simulations composed of different simulation models. From now on the different models, parts of the total model are called federates. In order to design a simulation composed of different federates on different computers it is necessary to connect them together and establish a communication protocol, this is done by HLA that clearly separate Simulation from Communication Process. As the HLA supports interoperability, the different federates must communicate with each other via the Run Time Infrastructure (RTI) according to the 10 HLA Rules.. The application of the HLA has many advantages since it offers interoperability, encourages reusability and makes it possible to use confident information in models without the necessity of being visible to other partners in the supply chain. In this way other partners don't have access to confident information and can't use it in a strategic way. The other partners will only see the results of simulation runs/steps and not the data behind the results. Because every partner now is more willing to use confidential information the total quality of a model of a supply chain increases and hence the benefits for the partners. Furthermore, because each partner builds it's own module, it is much easier to keep modules up-to-date both in term of data (i.e. directly from ERP). The University of Genoa was particularly involved in the Web Integrated Logistic Designer Project (WILD I & WILD II). These projects involve the development of a federation composed by simulators and dynamic programming systems (i.e. Nash Equilibrium Negotiation). The WILD Federation reproduces the supply chain and supports on-line the distribution among Suppliers, Main Contractors,

Proceedings 15th European Simulation Symposium
Alexander Verbraeck, Vlatka Hlupic (Eds.)
(c) SCS European Council / SCS Europe BVBA, 2003
ISBN 3-936150-28-1 (book) / 3-936150-29-X (CD)

Outsourcer and was successfully tested for an Supply Chain in the aerospace industry.
To make the HLA accessible also for Commercial Off of the Shelves Simulators (COTS) the HLA Operative Relay Using Sockets (HORUS™) has been developed. The HORUS™ manages the communication between the different simulation models and the HLA. To take advantage of interoperability and reusability existing federates, implemented in various languages, can be integrated.

High Level Architecture for Logistic

The High Level Architecture (HLA) is a standard framework that supports simulations composed of different simulation modules. Many complex simulations involve a combination of simulations of several different types of systems with different aspects of the total environment to be simulated. Unfortunately it is often necessary to make extensive modifications to adapt an existing simulation model so that it can be integrated into a new combined simulation (federation). In some cases it may prove easier to implement a completely new simulation of a system than to modify an existing one. In other words, traditional simulation models often lack two desirable properties: reusability and interoperability. Reusability means that modular simulation federates can be reused in different simulation scenarios and applications. Interoperability means that the reusable simulator can be combined with other federates into newly create exercises. without the need for re-coding. The HLA is an architecture that makes it possible for different modules (federates) to communicate with each other. A group of federates forms a federation.

HORUS™ for Integrating Arena in HLA

To make the HLA architecture more accessible to COTS, the MISS University of Genoa developed a Middleware called HORUSthat acts as a Delegate Simulator. A schematic overview of the HLA and HORUS is presented in figure 1.

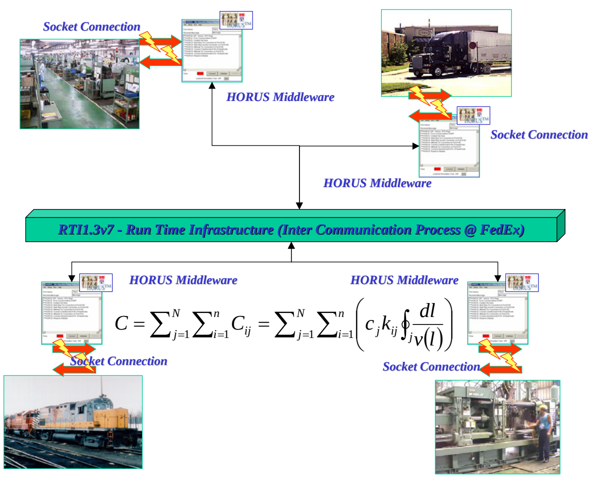

Figure 1 – HORUS Architecture

Each federate has it's own HORUS, in which a federate registers the attributes it would like to publish (to other federates) and subscribe to (from other federates). The HORUS is responsible for receiving information from HLA and sending information to federates via RTI. After it receives information, it passes it on to the HLA architecture. The HLA sends it back and the HORUS delivers it to a federate. The HORUS is used to make it easier to make a connection between the HLA and modules. One of the benefits is the ease to let the HLA know what kind of information a federate is interested (Declaration Management) in and what sort of information it wants to publish. In this way the RTI solves all the issues about the information routing. In this way it is possible to construct several parts of the model in different federates and to it in a simplified way. Possibilities for reusability are improved because the Building Block (Federates) have general processes, which can be used in other simulations without having to heavily re-code the model. When a federate contains too much specific processes, is not generally useful to be used in another simulation exercise. designed for general supply chains. Not only does this not improve reusability due to the greater complexity of a module, also more communication is needed to tailor the specific processes.

A Case Study

In order to practically demonstrate the proposed approach a Federation was designed in order to model a Supply Chain devoted to improve a Multi Drop Delivery Process. For this purpose Supply Chain was divided in three different modules. These are a Supplier, a Carrier and a Main Contractor Federate. In this project, cargo forms a bundle of loads. The Main Contractor sends orders to the Supplier. Besides ordering goods from the Supplier, the Main Contractor takes care of receiving cargo and has a terminal process to unload the transporters. When the Supplier has produced goods it sends a transportation request to the Carrier to transport the cargo. The requests have an attribute item, which indicates the number of loads (cargo) for a destination. The Supplier module has a production process (here the ordered goods are produced) and a terminal process (to load cargo on a transporter). The Carrier module provides the transporters and drivers. This module takes care of the transporters, drivers and of course the transport itself. The modular design makes it able to add extra modules (e.g. a maritime Carrier module) later on and to attach our modules to other modules. The HLA provides the possibility to connect Federates, but does, of course, not guarantee the usability of the information exchanged. In this project information is exchanged by means of formatted strings. In order to improve the data exchange String can be re-formatted in XML™ and furthermore parsed by the Simulator itself. To represent the objects described in this chapter, entities are used. In this way it is possible to design the supply chain in Arena™. The entities are rebuilt in the receiving modules based on strings composed of attributes. The entities chosen are orders, requests, cargo and transporters. The following general attributes, which are necessary for a supply chain, are chosen:

- Origin: is unique identifier of the pick up location.
- Destination: is the unique identifier of the delivery location.
- Transporter: is the unique code that identify the Carrier into the System
- Items: is the dimension of the batch that is shipped in this delivery.
- Kind: is the identification of the typology of the cargo (i.e. bulk, parcels, gas, etc.), it is used to identify the best suitable vehicle.
- Id: is the unique id for every order, it is used for the cross reference on the Federation.
- Time_Delivered: is the time when cargo is delivered to the Main Contractor

Transporters have some general attributes and some extra attributes that have to do with the architecture of the modules. For the transporters the following extra attributes are chosen: Last_Destination, Status and Costs. Last_Destination indicates the destination a transporter last visited. Status indicates whether a transporter is in use or not and costs are the transportation costs for the cargo. These are the attributes specific enough for a supply chain and general enough to couple modules to other modules as other modules designed for a supply chain will use these kind of attributes.

In the supply chain there are three terminals and several destinations taken into account. The terminals serve as home stations for the transporters. The terminals also have the cargo the Supplier produces in stock. Unlike an hub-and-spoke network is there no transport of cargo to the terminals (hubs). There is only one way transport of cargo from the terminals to the destinations. So the terminals act as distribution centers. Main Contractor serve as a system that orders goods for multiple locations (see Figure 2). The Supplier serves as a system that resembles the production process for all orders the Main Contractor sends to the Supplier.

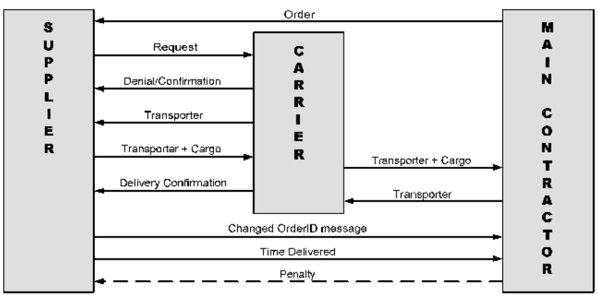

Figure 2: Federation Architecture

The Carrier serves as a system that picks up cargo from the terminals and transports it to the different eleven destinations (the locations). From the terminals the cargo is distributed to the different destinations. The Main Contractor sends orders to the Supplier. These orders have one of the terminals as origin. The destination can be every location, except its own origin.

In the Supplier four objects arrive, namely orders from the Main Contractor and empty transporters, a delivery confirmation and an answer to the transportation request from the Carrier. The Supplier also sends some objects: loaded transporters, transportation requests and time delivered.

Suppliers' Logic

When the Supplier receives orders from the Main Contractor, it starts to produce the ordered goods.

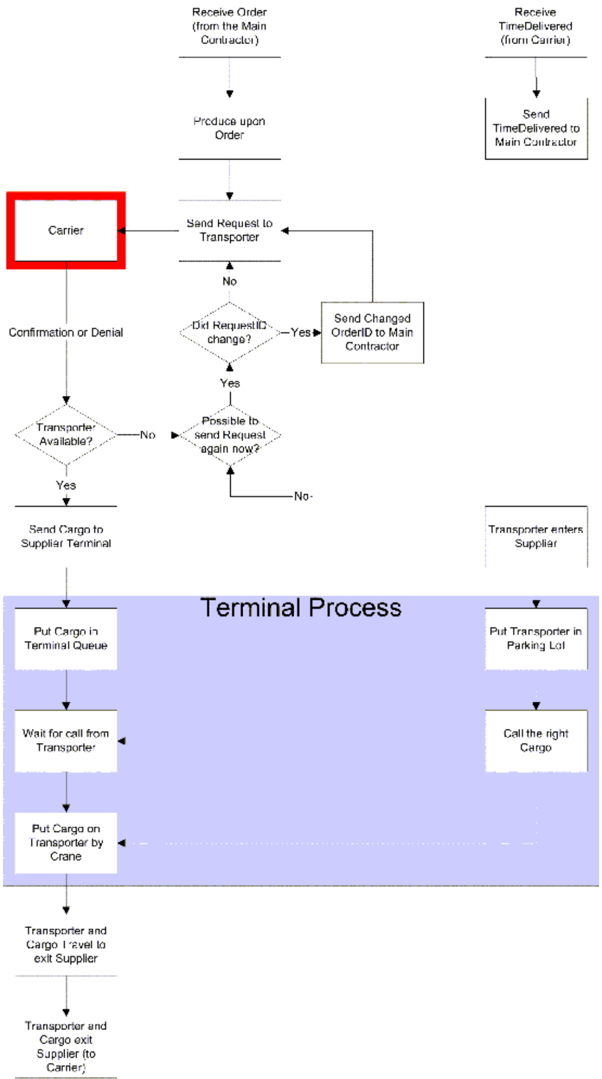

Figure 3 – The Suppliers' Logic Diagram

When the production is finished, requests are created. A request represents a number of loads (cargo) for a destination. These are sent to the Carrier. After a while the Carrier sends a message whether it is possible or not to transport this cargo. So the carrier decides whether it is possible to transport cargo to a destination (see Figure 3). Depending on the answer of the Carrier the Supplier sends the cargo to the Supplier terminal or, when the Carrier gives a negative answer, the Supplier sends the request for the same cargo again in the next time step. When a request can not be sent immediately (due to the unavailability of a transporter), the Supplier keeps on trying to send denied requests to the Carrier. Whether denied requests are sent immediately depends on the number of other requests waiting for transportation. The Supplier decides whether it is possible to immediately sent denied requests again or not. When a transporter arrives at the Supplier it goes to a central parking lot. After a while it calls its cargo, which it has to transport, from the terminal queue. This is done when the crane in the terminal is free, because there are no other transporters to serve. After the transporter has called its cargo, it travels to the crane. The called cargo also goes to the crane. The crane puts the cargo on the transporter. When the transporter is fully loaded, it leaves the terminal and travels, with its cargo, to the exit of the Supplier and exits. Its next event is entering the Carrier. In the Supplier, beside empty transporters, also delivery confirmations arrive from the Carrier. When a transporter has delivered its cargo and returns to the Carrier it sends a delivery confirmation to the Supplier.

Carriers' Logic

In the Carrier three objects arrive, namely requests and transporters with cargo from the Supplier and empty transporters from the Main Contractor. The Carrier also sends messages, confirmations or denials and delivery confirmations to the Supplier (See Figure 4).

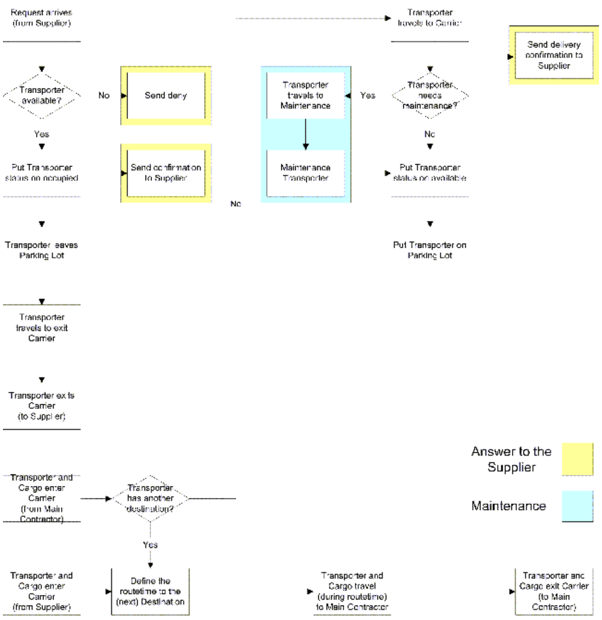

Figure 4: Carriers'Logic Diagram

When a request arrives from the Supplier, the Carrier checks whether a transporter is available or not for the requested destination. Here the Carrier decides whether requests for transportation are denied or approved. If a transporter is available the status is set on occupied. If no transporter is available the Carrier sends a negative answer (denial) to the Supplier. Otherwise a positive answer (confirmation) is send to the Supplier.

When in the Carrier transporters arrive with their cargo from the Supplier, they travel (during their RouteTime) in the Carrier to the Carrier exit and enter the Main Contractor.

When transporters arrive at the Carrier from the Main Contractor first of all a delivery confirmation is sent to the Supplier. Next the decision is made if they have another destination besides returning to the nearest terminal. If another destination is found, the transporter travels to its this destination. If there is no destination it has to visit, it travels to its nearest terminal. At the same time the decision is made whether or not the transporter needs some maintenance. If it needs some maintenance, the transporter travels to the maintenance. Otherwise the transporter travels to the parking lot corresponding to its nearest terminal. When a transporter arrives at the maintenance, it undergoes repair for a certain amount of time. After this period the transporter travels to the parking lot.

Main Contractors' Logic

The Main contractor sends two objects. These are orders and transporters (with or without cargo). The orders are created and sent to the Supplier. When the orders are sent, their time and some other characteristics are written in a sheet and an expected time of arrival of the goods is estimated (see Figure 5). In the Main Contractor three objects arrive, namely transporters with their cargo from the Carrier, a delivery confirmation and a Changed OrderID message from the Supplier.

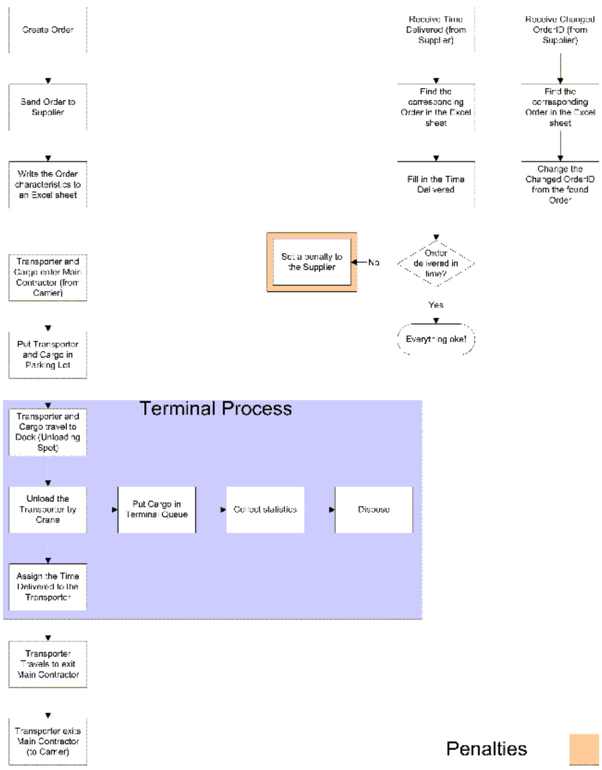

Figure 5: Main Contractors' Logic Diagram

In the Main Contractor's transporters arrive from the Carrier carrying its cargo. When they arrive at the Main Contractor, they go to a central Parking Lot, where they will wait. When the crane is available, a transporter travels to the terminal. After a transporter arrives at the terminal, a crane unloads it. The cargo is put in the Main Contractor’s terminal, where some statistics are collected and after that the cargo is disposed. As soon as a transporter is unloaded, its Time_Delivered attribute is set and it travels to the Main Contractor’s exit, exits the Main Contractor and enters the Carrier. When the cargo is delivered, the transporter sends a delivery confirmation, based on the Time_Delivered, to the Supplier after it entered the Carrier. After a while the Main Contractor receives a delivery confirmation from the Supplier including the OrderID, so the Main Contractor can match this with its original send orders. Together with this time and the expected delivery time, a calculation is made whether a penalty should be given to the Supplier. When a Changed OrderID message is received in the Main Contractor, this is matched with the original orders and the OrderID of the corresponding order(s) is changed in the changed OrderID. This is necessary regarding the delivery confirmation in a later stadium. When two orders are combined into one order, one of the OrderID's is changed. To make it possible to know in a later stadium the order (with the changed OrderID) is delivered it is necessary to change its OrderID into the new one.

Terminals’ Logic

As the terminal process is depending on two objects, namely cargo and transporters, both must be available for the terminal process. For this matter it is not enough to just model the terminal process as a stochastic delay. As there are different modules, the terminal processes are assigned to modules. As cargo can not transport itself in reality, transporters are moved to other modules. For this reason the terminal processes are assigned to the Supplier and the Main Contractor.

The Carrier receives requests from the Supplier and sends transporters to the Supplier. The Supplier receives them on its own internal terminal queue. After loading, the Supplier returns the transporters and their cargo to the Carrier. The same goes for the relation between the Carrier and the Main Contractor. In this situation it concerns an unloading process.

VV&A: Learned Lessons

Verification is the process of determining that a model implementation accurately represents the developer’s conceptual description and specifications. Validation is the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model. Due to the exploring character of this project, many problems were already found and solved during the design phase. Verification of distributed models is more difficult than verification of non-distributed models (ignoring differences in logic and size of the models). This because when something goes not as expected, the source of the problem is hard to find. It is possible that a problem in one module is caused by an error in another module. In this way it is not possible to isolate problems easily and in many cases multiple (long lasting) runs are necessary to find a problem and to solve it.

Conclusions

Experimental Campaign is under development and preliminary results show great potential of the outlined methodology.

Due to the extreme long runtime that experiments take an replicated half-fractional factorial 2^{2-1}x 2 design was used, the obtained Response Surface proven that a such federation could be used in order to provide a meta model able to improve the performance of a Production Planner.

Commercial stand alone modeling software used to build HLA Supply Chain Federates have shown long simulation run time that can reduce significantly the use of the simulator on the other hand the use of regression meta models must be considered as a performing alternative.

References

1. Bruzzone A., Mosca R., Revetria R. (2001) "Supply Chain Management over the web in Aerospace Industry by using Simulation: WILD", Proceedings of Virtuality2001, Turin
2. Bruzzone A., Roberto Mosca, Flavio Tonelli, R.Revetria, Vigano' G., Diglio G. (2001) "Advanced Issues in Distributed Verification and Validation Process for Supply Chain
3. Bruzzone A.G, Mosca R., Revetria R., Tonelli F. (2002) "Sistemi Integrati di Gestione Avanzata per la Quick Response della Catena di Fornitura", Proceedings of XXIX Convegno Nazionale ANIMP, Sorrento, Italy October
4. Bruzzone A.G., Giribone P., Revetria R. (2002) "Genetic Algorithms and Simulation for Aftersales Supply Chain Re-Engineering Process", Proceedings of MIC2002, Innsbruck, February 18-21
5. Bruzzone A.G., Giribone P., Revetria R., Simeoni S. (2001) "Potential of Artificial Intelligence Techniques and Simulation in Improving Container Terminal Performances", Proceedings of POMS2001, Orlando, March 31 April 2
6. Bruzzone A.G., Mosca R., Revetria R. (2001). "Gestione Integrata di Sistemi Produttivi Interagenti: Metodi Quantitativi Avanzati per la Quick Response", DIP Genova, Italy, ISBN: 88-900732-0-9
7. Bruzzone A.G., Mosca R., Revetria R. (2002) "Cooperation in Maritime Training Process using Virtual Reality Based and HLA Compliant Simulation", Proceedings of XVIII International Port Conference, Alexandria Egypt, January 27-29
8. Bruzzone A.G., Mosca R., Revetria R. (2002) "Gestione della Supply Chain Mediante Federazione di Simulatori Interagenti: Compendium", DIP University of Genoa, Genoa, December ISBN 88-900732-1-7

9. Bruzzone A.G., Mosca R., Revetria R. (2002) "Supply Chain Management Dynamic Negotiation using Web Integrated Logistics Designer (WILD II)", Proceedings of MAS2002,
10. Bruzzone A.G., Mosca R., Revetria R. (2002) "Web Integrated Logistics Designer and Intelligent Control for Supply Chain Management", Proceedings of Summer Computer Simulation Conference 2002, San Diego, July
11. Bruzzone Agostino, Mosca R., Revetria R. (2001) "Web Integrated Logistics Designer: A HLA Ferderation Devoted to Supply Chain Management", Proceedings of Summer Computer Simulation Conference, Orlando, July 15-19
12. Giribone P., Revetria R., Mantero E., Diglio G., Vigano'G., (2000) "A General Purpose System for Supporting Transportation System Re-Engineering", Proceedings of HMS2000, Portofino, October 5-7
13. Revetria R., Blomjous P.E.J.N.,. Van Houten S.P.A (2001)“Simulating Supply Chain Logistics with Arena™ and The High Level Architecture” DIP Tech. Report, Savona January
14. Revetria R., Tucci M. (2001) “Different Approaches in Making Simulation Languages Compliant with HLA Specification” Proceedings of the Summer Computer Simulation Conference, Orlando FL, July 14-19