IBIMA Publishing
Communications of International Proceedings
https://ibimapublishing.com/p-articles/44AI/2024/4454224/
Vol. 2024 (22), Article ID 4454224

Mathematical Model of Communication of Multiple Simulators in An Environment Based on The HLA Standard*

Krzysztof PANUFNIK and Tadeusz NOWICKI

Faculty of Cybernetics, Military University of Technology,
00-908 Warsaw, Kaliskiego 2 Street, Poland

Correspondence should be addressed to: Krzysztof PANUFNIK, kpanufnik@gmail.com

* Presented at the 44th IBIMA International Conference, 27-28 November 2024 Granada, Spain

Abstract

The paper presents a mathematical model of the communication process of simulators connected in the HLA architecture (high-level architecture). Simulators connected in this architecture, called federates, often use common data, the values of which can change. Information about changes is sent to other federates interested in these changes. The problem of choosing a method of notifying other federates about the changes that have occurred arises. Communication usually places a heavy burden on the network with federates. The motivation for this work was to develop a strategy for broadcasting information about changes that causes the lowest communication load in the network. There are few methods presented in the literature that address this problem. Moreover, they are not effective enough. The paper presents a mathematical (stochastic) model of communication processes between federates and criteria for assessing the efficiency and effectiveness of these processes. The next step will be to develop a simulation model and experimental research methodology for assessing the effectiveness of communication control methods in the HLA architecture.

Keywords: simulation, standard HLA, data distribution service, data communication management

Introduction

In distributed simulation, one of the most important problems is the time synchronization of the simulation processes, called Time Management. The problem is to synchronize the changes in simulation time, because the computational processes are performed at different speeds in individual simulators. The growing size of simulation experiments has made the exchange of data within the experiments a significant problem. Simulations often reach the size of tens or hundreds of thousands of independent objects, distributed between many simulators. In the case of simulations of this size, it is extremely important for the speed of execution to limit the amount of information that is transmitted via the computer network.

Over the years, many dedicated simulation protocols and standards have been created to support solving the above problems. At the beginning of the 2000s, a new standard for connecting simulators was implemented, called High Level Architecture (HLA) [1]. HLA has now become the dominant standard for connecting simulators. Simulators that create a simulation environment are called federates. Although HLA is a description of the simulation architecture, it assumes the existence of a central software called Runtime Infrastructure (RTI).

The HLA standard assumes that a common data model is used, described by the simulation object model (SOM)

Cite this Article as: Krzysztof PANUFNIK and Tadeusz NOWICKI, Vol. 2024 (22) “Mathematical Model of Communication of Multiple Simulators in An Environment Based on The HLA Standard " Communications of International Proceedings, Vol. 2024 (22), Article ID 4454224, https://doi.org/10.5171/2024.4454224

[2]. SOM is a formal description of simulation objects and their attributes and interactions that can be distributed and/or received by a federate. The set of all objects and interactions described in the SOM, together with additional information such as data representation, forms a federation object model (FOM). A group of federates, with associated SOM models, connected to the RTI, together with the FOM model, is called a Federation. Objects have attributes that reflect their state and are important from the point of view of the simulation experiment.

The values of some attributes may change frequently, others less frequently, and still others do not change. Management of activities related to objects and interactions used during simulation is possible thanks to the object management module (OM). The federate that owns an object attribute, when making changes to it, notifies other interested federates via RTI. Transferring ownership of an attribute is possible using the RTI module in the scope of the so-called ownership management.

HLA provides a mechanism for managing interests in the form of a data distribution management (DDM) service. While the OM service allows the expression of interest in data at the object class or interaction level, DDM allows the expression of interest in data at the object attribute level.

In simulation processes based on HLA, a data space is defined, which is usually intensively modified during a simulation experiment. The entire data space consists of attributes of all simulation objects. This space can be divided into separate subsets called regions, which will be called data subareas in the following. The division of the space depends, among other things, on the type of simulation experiment and the needs of the federates participating in it.

HLA provides the possibility of limiting the information about changes received by the federate. This is done thanks to the federate declaring the data subareas about the changes of which the federate wants to be informed. Modification of a data subareas by a federate means that the modified data must be sent to all federates interested in those subareas. It is assumed that changes can be sent by the federate using one of two types of communication channels: one-to-one or many-to-many.

Using a one-to-one channel when many recipients are interested in each data subarea results in the necessity to send a given change multiple time. The situation in which the federate is forced to send information about a change multiple times is called the burden of the sending federate.

From the point of view of the sending federate, the disadvantage of the one-to-one channel can be eliminated by using a many-to-many channel. By using such a channel, the change is sent once and then it will be delivered to all recipients who have declared their willingness to listen to this channel. If the many-to-many channel is used to send multiple subareas (to one or more recipients), a situation may arise when not all listeners of the channel data may be interested in changes in all subareas that are sent using such a channel. The situation in which the federate receives information about a change for a subarea in which it was not interested is called the burden of the receiving federate.

Due to the limited number of many-to-many channels, the potentially large number of possible sub-areas and the different interests of the federates, it is assumed that the number of many-to-many channels available for use is insufficient for complex experiments. As a result, a simulation experiment may lead to a situation where there is no such assignment of individual federates to channels that would avoid excessive load on the federates. The aim of the research was to develop a new and effective method for selecting a communication strategy for simulators in an HLA-based environment. The selection of a communication strategy has a significant impact on the performance of a complex simulation experiment. The strategy affects the load of federates and, consequently, the time between the generation of a change by a federate and the update of information by all federates interested in each change. The use of a communication optimization method between federates connected via HLA will allow for the improvement of one or more of the following parameters: the load of sending federates, the load of receiving federates or the average time of sending information between federates. The most advanced results on this subject published in the literature are the works [3] and [4].

To develop a new and more effective method for the communication strategy between simulators, it was necessary to develop a mathematical model of federate communication in the HLA environment and to formulate the task of federate communication optimization. The article contains a mathematical model of communication between multiple simulators in an environment based on the HLA standard.

Mathematical description of model parameters

Let's assume that the time period for which the communication strategy is established is:

$$t_{asn}. \tag{1}$$

After this time, it is possible to continue using the selected strategy or change it to a new one in case of a significant change in the simulation parameter values. The set of federate numbers participating in the simulation experiment is marked as:

$$\mathbb{F} = \{1,2,\ldots,f,\ldots,F\}. \tag{2}$$

Each federate can declare publication or subscription regions. After the DDM mechanism is started, intersections between publication and subscription regions belonging to different federates are found. From the point of view of communication between federates, only those objects that belong to the intersections of publication and subscription regions are important. These intersections are a set of disjoint subareas of data with numbers from the set:

$$\mathbb{O} = \{1,2,\ldots,o,\ldots,O\}. \tag{3}$$

Based on the intersections between the publishing and subscription regions, it is possible to determine which federates should receive changes, in each subarea, from federates that change a given subarea. The matrix defining which federates should receive changes in subareas is of the form:

$$\mathbb{S} = \left[s_{f_1,o,f_2}\right]_{\mathbb{F}\times\mathbb{O}\times\mathbb{F}} \ , \ \ s_{f_1,o,f_2} \in \{0,1\} \tag{4}$$

where s_{f_1,o,f_2} means that federate **f₁** will send changes in subarea **o** to federate **f₂** if the value of s_{f_1,o,f_2} is 1 and will not send changes if the value of s_{f_1,o,f_2} is 0. A given data subarea can be modified by more than one federate. In case a federate modifies a subarea, it is interested in, it will not send the changes in that subarea to itself.

Thus:

$$\forall_{s_{f_1,o,f_2}} f_1 = f_2 \Rightarrow s_{f_1,o,f_2} = 0, \tag{5}$$

It is assumed that changes in subareas are generated with a given probability, whereby for each subarea and federate the probability distribution may be different. Random variables describing the times between subsequent changes of the subareas, performed by the federate are defined in the matrix:

$$\mathbb{T} = \left[t_{f,o}\right]_{F\times O}, \tag{6}$$

where $t_{f,o}$ denotes a random variable describing the time between subsequent changes in area **o** made by federate **f**. For any random variable, one can determine the renewal function of the stochastic process describing the expected number of changes made by the federate in the subarea by time **t**. Thus, the following functions are defined:

$$H_{f,o}(t), f \in \mathbb{F}, o \in \mathbb{O}, \tag{7}$$

where $H_{f,o}(t)$ denotes the expected number of changes made by federate **f** by time **t** in subarea **o**.

The size of the change sent depends on the event that took place in the simulator and the object related to that event. Therefore, it is assumed that the size of the change is random. Random variables describing the size of the change, counted e.g. in bytes, made by the federate in the subarea are defined in the matrix:

$$\mathbb{V} = \left[v_{f,o}\right]_{F\times O}, \tag{8}$$

where $v_{f,o}$ denotes a random variable describing the size of the change made by the federate **f** in the subarea **o**. Due to the changing intensity of the simulation or additional processes running on the device on which the federate is running, the resources available to the federate process change over time. Consequently, the time needed to send or receive a message by the federate may be different at different moments of the experiment. These times are described by random variables. The random variable describing the time needed to send a message by the federate is denoted by:

$$ts_i, \ i \in \mathbb{F}, \tag{9}$$

The random variable describing the time needed to receive a message by a federate is denoted by:

$$tr_i, \ i \in \mathbb{F}, \tag{10}$$

The structure of the computer network and the location of the federates in it affect the throughput of the link between the federates. The throughput of the link between a pair of federates is defined in the matrix:

$$\mathbb{C} = \left[c_{i,j}\right]_{F\times F}, c_{i,j} \in \mathbb{R}_+, \tag{11}$$

where $c_{i,j}$ denotes the link capacity between federates **i** and **j**.

There are two types of communication channels available to federates. One-to-one and many-to-many channels. The set of all communication channel numbers is denoted:

$$\mathbb{M} = \{1,2,\ldots,m,\ldots,M\}, \tag{12}$$

where the number of many-to-many channels is denoted as:

$$r, r < M, \tag{13}$$

Decision variable definition

The decision variable in the problem of selecting the simulators' communication strategy is defined by the matrix:

$$\mathbb{X}\left[x_{f,o,m,t}\right]_{F\times O\times M\times 2}, x_{f,o,m,t} \in \{0,1\}, \tag{14}$$

where $x_{f,o,m,0} = 1$ means that federate **f** is to send (t=0) information about changes in area **o** using channel **m**. Similarly, $x_{f,o,m,0} = 0$ means that federate f is to receive (t=1) information about changes in area **o** using channel **m**.

Defining constraints for the decision variable

A connection is a triplet: a sending federate, the size, and the set of receiving federates. A connection describes the stream of changes generated by a federate in each subarea, together with the set of federates interested in the changes in the given region. The size of the connection is understood as the expected number of changes generated by a federate. Information about data flows between federates can be represented as a multigraph, whose nodes are federates and edges are the connection. An example of a connection graph is shown in Figure 1. The graph represents the transmission of changes in 4 subareas by 3 federates. Federate 1 sends changes in two subareas: 1 with weight 20 to federates 2, 3 and 4 and 2 with weight 5 to federates 2 and 3. Federates 3 and 4 change one sub-area each, respectively subarea 3 with weight 5, which is sent to federate 2, and subarea 4 with weight 15, whose recipients are federates 2 and 3.

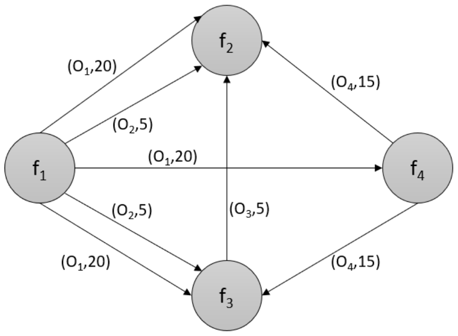

Fig. 1. Connection graph containing information about which federate sends changes in each subarea and who are the recipients of these changes.

Combining the effects of the DDM scheme and the federate parameters, we get a connection graph. This graph clearly indicates which subareas are to be sent to the interested federates. For each pair of federates, if there is a requirement for federate 1 to send changes in a subarea to federate 2, it is expressed as:

$$\forall_{f_1,f_2\in\mathbb{F},o\in\mathbb{O}}\; s_{f_1,o,f_2} = 1, \tag{15}$$

then at the same time there is a channel through which federate 1 sends changes in the subarea, and federate 2 receives these changes in the subarea:

$$\exists_{m\in\mathbb{M}}\left(x_{f_1,o,m,0} = 1 \wedge x_{f_2,o,m,1} = 1\right), \tag{16}$$

From the connection graph, it follows that there is no subarea that is changed by a federate, but in which no other federate is interested. Similarly, it follows that there is no federate that is interested in a subarea that no other federate changes. Therefore, there is no channel that a federate uses to send (receive) changes in a subarea, for which there is no federate that uses the channel to receive (send) changes in the subarea. For each channel, then, the following holds:

$$\forall_{m\in\mathbb{M}}\left(\exists_{f_1\in\mathbb{F}}x_{f_1,o,m,0}=1 \Leftrightarrow \exists_{f_2\in\mathbb{F}}x_{f_2,o,m,1}=1\right). \tag{17}$$

Number of channels that have more than one sending or receiving federate does not exceed the number of many-to-many channels:

$$\sum_{m\in\mathbb{M}}\begin{bmatrix} \left|\left\{f\colon f\in\mathbb{F}\wedge\exists_{o\in\mathbb{O}}x_{f,o,m,0}=1\right\}\right|>1 \\ \vee\left|\left\{f\colon f\in\mathbb{F}\wedge\exists_{o\in\mathbb{O}}x_{f,o,m,1}=1\right\}\right|>1 \end{bmatrix}\leq r \tag{18}$$

Criteria for optimization problem

The choice of communication strategy affects the load of sending and receiving federates and the time of sending messages. To assess the quality of the solution, three criterion functions g_1, g_2 and g_3 are introduced. ***The function g_1 describes the excess load of sending federates*** understood as the difference between the number of sent changes and the number of changes sent by time t that the federate would have send if each change were sent only once.

In an ideal situation for the sending federate, it would have to send each change only once:

$$g_1''(t,f_1)=\sum_{o\in\mathbb{O}}\left[\exists f_2\in\mathbb{F}\Rightarrow s_{f_1,o,f_2}=1\right]\cdot H_{f_1,o}(t), \tag{19}$$

In practice, due to the limited number of many-to-many channels, the strategy selection algorithm may cause that a change with multiple recipients must be sent over a one-to-one channel. The number of sent changes by a federate is calculated as the sum of the expected number of changes in the subarea, for each communication channel:

$$g_1'(\mathbb{X},t,f_1)=\sum_{m\in\mathbb{M}}\sum_{o\in\mathbb{O}}x_{f_1,o,m,0}\cdot H_{f_1,o}(t), \tag{20}$$

Finally, the total excess load of sending federates can be calculated as the sum of the loads of the individual federates:

$$g_1(\mathbb{X},t)=\sum_{f\in\mathbb{F}}\left(g_1'(\mathbb{X},t,f)-g_1''(t,f)\right), \tag{21}$$

The function g_2 describes the excess load on receiving federates, defined as the difference between the number of received changes and the number of changes that the federates would have received if they had only received the changes they were interested in. In the ideal situation, the receiving federate would have received only the changes it was interested in:

$$g_2''(t,f_1)=\sum_{o\in\mathbb{O}}\sum_{f_2\in\mathbb{F}\backslash f_1}s_{f_2,o,f_1}\cdot H_{f_2,o}(t), \tag{22}$$

In practice, it may happen that a many-to-many channel is used to send changes to subareas that the federate was not interested in. The number of changes received by a federate is calculated as the sum of the expected number of changes to subareas sent over channels that the federate joined:

$$g_2'(\mathbb{X},t,f_1)=\sum_{m\in\mathbb{M}}\sum_{o\in\mathbb{O}}\sum_{f_2\in\mathbb{F}\backslash f_1}x_{f_2,o,m,0}\cdot x_{f_1,o,m,1}\cdot H_{f_2,o}(t), \tag{23}$$

Finally, the total excess load on receiving federates can be calculated as the sum of the loads on the individual federates:

$$g_2(\mathbb{X},t)=\sum_{f\in\mathbb{F}}\left(g_2'(\mathbb{X},t,f)-g_2''(t,f)\right), \tag{24}$$

The g_3 function defines the system load, understood as an upper estimate of the time to send changes, in a time interval of length t, between the most loaded pair of federates. This function is introduced due to the difficulty in estimating the time to send a message, which depends on the time spent in the queues of the sending and receiving federates. Analytical determination of these values would be possible only for selected probability distributions used in the system. For the sending federate, the necessary time to send all the messages that the federate must send to the other federates can be calculated:

$$g_1{}'(\mathbb{X},t,f_1)\cdot E\left\{ts_{f_1}\right\}, \tag{25}$$

The time required to transfer all changes between a pair of federates is expressed as the number of bytes to be transferred between federates divided by the bandwidth of the link between them:

$$\frac{g_3{'}(\mathbb{X},t,f_1,f_2)}{c_{f_1,f_2}}, \tag{26}$$

where the number of bytes can be calculated using the following formula:

$$g_3{'}(\mathbb{X},t,f_1,f_2) = \sum_{m\in\mathbb{M}} \sum_{o\in\mathbb{O}} x_{f_1,o,m,0} \cdot x_{f_2,o,m,1} \cdot H_{f_1,o}(t) \cdot E\{v_{f_1,o}\}, \tag{27}$$

For a receiving federate, the time required to receive all messages from other federates can be calculated:

$$g_2{'}(\mathbb{X},t,f_2) \cdot E\{tr_{f_2}\}, \tag{28}$$

The system load can be estimated by finding the pair of federates for which the sum of the above three times is the largest:

$$\begin{aligned} g_3(\mathbb{X},t) = max\Bigg\{ & g_1{'}(\mathbb{X},t,f_1) \cdot E\{ts_f\} + \frac{g_3{'}(\mathbb{X},t,f_1,f_2)}{c_{f_1,f_2}} \\ & + g_2{'}(\mathbb{X},t,f_2) \cdot E\{tr_{f_2}\} : f_1,f_2 \in \mathbb{F}, f_1 \neq f_2 \Bigg\} \end{aligned} \tag{29}$$

The second component of the above sum assumes that the transmission of each change takes place when no other change is being transmitted between the pair of federates. If the transmission in each network took place only between one pair of federates, it can be assumed that this situation would be equivalent to the simultaneous transmission of all changes (assuming that such a total transmission does not saturate the link). In the case of simultaneous transmission, each change would use only a part of the available bandwidth. Therefore, the transmission of a single change would take correspondingly longer, but the total time should remain unchanged. In practice, information is exchanged simultaneously between different federates using different channels. The influence of communication of one pair of federates on the other federates strongly depends on the structure of the communication network used. The same applies to the influence of using different communication channels.

Since it would be difficult to consider these aspects of reality, the model assumes a simplified network model in which:

- transmission of one change does not affect the transmission time of other changes,
- use of a many-to-many channel instead of a one-to-one channel brings benefits to the federate directly proportional to the number of recipients of this channel.

HLA is a standard that does not specify how RTI software should be created. Among the implementations, we can distinguish centralized and distributed RTI [5]. In the case of centralized RTI, the simulation process is managed by a single RTI component, which can become a bottleneck. In the case of distributed RTI, individual HLA services are deployed in separate network nodes, which ensures load distribution. Regardless of the RTI type, it can be assumed that the DDM service only determines the channels to be used for communication and announces this information to the federates. Then, the federates send information about changes in objects using these channels, without involving the RTI process itself. Otherwise, for complex experiments, all network traffic would have to pass through the RTI node, which would become a bottleneck. Therefore, in this work, model was created with assumption that RTI only serves to determine communication channels, and the transmission of information about changes is performed without RTI.

Conclusions

The paper describes a mathematical model of communication processes between simulators operating in the HLA architecture. This model is needed to analyze the characteristics of the simulation, including the communication load. The proposed mathematical model of communication in HLA differs significantly from the previous models described in [3] and [4], including the introduction of random elements. The number of changes and the time of transmission of the change between a pair of federates were constant values. In the proposed model, the number of changes is described by a random variable, and the transmission time depends on the size of the change, which is random, and the bandwidth of the link between the pair of federates. Similarly, the times of sending and receiving the change are described by random variables, while in the previous solutions they were constant. Due to the introduction of random elements, it was necessary to develop new criteria for evaluating the solution. The introduction of a random element requires building a simulator to test the communication strategy, because analytical methods of finding a solution would allow investigating only a narrow subset of possible cases. As a result, the performed calculations would not be reliable and would not reflect the real cases of using HLA.

Acknowledgements

The work was financed by the Military University of Technology in Warsaw, Poland as part of the project No. UGB 22-701/2024.

Bibliography

- Topçu O., Oğuztüzün H. (2017). Guide to Distributed Simulation with HLA, Springer International Publishing, Simulation Foundations, Methods and Applications, doi.org/10.1007/978-3-319-61267-6.
- 1516.2-2010 - IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Object Model Template (OMT) Specification (2010).
- Morse K., Zyda M. (2002). Multicast grouping for data distribution management. Simulation Practice and Theory, vol. 9, issues 3-4, pp. 121–141, doi.org/10.1016/S0928-4869(01)00054-4.
- Wang J., Zheng T. (2014). A hybrid multicast–unicast assignment approach for data. Simulation Modelling Practice and Theory, vol. 40, pp. 39–63, doi.org/10.1016/j.simpat.2013.08.005.
- Akram A., Sarfraz M. S., Shoaib U. (2019). HLA Run Time Infrastructure: A Comparative Study. Mehran University Research Journal of Engineering and Technology, vol. 38, issue 4, pp. 961-972, DOI:10.22581/muet1982.1904.09.