Easy Communication Approach for Data Exchange in Distributed Simulation Environment

ARTIS AIZSTRAUTS, EGILS GINTERS, DACE AIZSTRAUTA
Sociotechnical Systems Engineering Institute
Vidzeme University of Applied Sciences
Cesu Street 4, Valmiera LV-4200
LATVIA
{artis.silins, egils.ginters}@va.lv http://www.socsimnet.com

Abstract: - The development of simulation enables the researchers to explore more sophisticated problems at a more detailed level. At the same time this implies a necessity to use several models, or even different modelling tools. Different modelling approaches (multi agent systems, discrete events simulation, system dynamics, etc.) usually envisage the use of different modelling tools (software). This makes the issue of their mutual communication a central concern to the researcher. There are several solutions, but they are complicated and not suitable for people without specific knowledge. The aim of the article is introduction to new and easy communication environment which would be used to substitute HLA and other more expensive tools.

Key-Words: - Simulation, HLA, Distributed simulation, Easy communication

1 Introduction

At the level of decision making the consolidation of all data is necessary. Therefore, the demand for the communication environment which ensures the information exchange with appropriate efficiency is important. Many solutions exist, but HLA [10] can be nominated as more popular one. However, the high costs of the commercial versions promote for new ways searching and using of the distributed environments provided for software modules communication.

Meanwhile the variety of simulation tools, including the development of different user-friendly simulation tools, enables professionals of various fields to use them in their everyday routine. Therefore mechanisms for simulation models communication should be useful for these professionals, they should be unsophisticated and user-friendly, that do not require any specific knowledge or skills.

Hence this paper describes a solution that would give the opportunity to ensure the communication between modelling tools or distributed models without HLA communication architecture at the same time being unsophisticated and appropriate for users without extensive technical skills. It is primarily based on the "Easy communication environment for distributed simulation" [1]. Instead of HLA this paper concentrates on a newly developed inter-adapter communication mechanism.

2 How to Access the High Level Architecture (HLA)

High Level Architecture (HLA) [11] is a concept of the architecture for distributed simulation systems. HLA ensures interoperability and reuses among simulations. It consists of rules that separate parts of distributed simulation model (federates) must follow to achieve proper interaction during a federation execution.

Specification, which provides interface to the Run-Time Infrastructure, which can be distributed and ties together federates during model execution. The distributed time management can be done, because all federates' nodes directly undertake synchronization roles. Therefore, the total simulation is takes less time and the system is safer, unfortunately, implementation is very expensive, complex and laborious.

The wide variety of HLA functionality is rarely needed and only applicable to specific simulations, therefore one might ask, do we really need to use HLA? After investigating different possible options for data exchange among different models and simulation tools, the authors elaborated a pilot-solution – easy communication environment that facilitates access to the HLA environment [1]. Communication was possible locally (within one computer) as well as globally (among several computers). For local communication this solution used communication adapter – independent software

– that was on the same computer as the simulation tool. In case of global communication, the solution used HLA architecture as well as the adapter(s).
For this solution to become real, the simulation tools need to have a feature to create extensions. These extensions provide data exchange between the model and the "outside". As one cannot be sure, that the model is able to ensure live instance communication channel, therefore data storage for data storing was created. Simulation tool with the help of extension received data from the data storage. Also the transfer of data was ensured by the extension – the data were transferred to communication adapter, and later on were transferred to other communication adapter with the help of HLA. It should be noted that this mechanism did not use time synchronization that is realised by HLA to ensure that in whole federation there would be one time and separate federer would not hasten. Basically HLA was used as a communication adapter switch that allows them to communicate with each other as well as it provided a list with possible interactions. Considering the abovementioned, there is no need to "force" the usage of HLA as it is rather powerful and capable communication architecture, nonetheless it is not worth wile to use it for comparatively easy tasks.

3 New Approach to Easy Communication

3.1 Fundamentals

The basic principles of the proposed mechanism is similar to one described in "Easy communication environment for distributed simulation pub."[1], only instead of HLA another communication mechanism is used for data transfer to other communication adapters. Changes have been made also to the structure of data that is being transmitted between simulation tool and communication adapter, as well as among several communication adapters. The main elements of this mechanism are the following: *Models (Simulation tools)*, *Communication library, Communication adapter, Data storage* and *Communication gateway.*

In order to be able to function in a communicative system, *Simulation tools* have to be able to communicate with itself or different simulation tool. In this article strictly focus only one of these versions, namely, where simulation tools can perform communication with other software (adapter). In most cases this function is performed with some additional plug-ins, extensions or libraries. For example, NetLogo, that is a multi agent-modelling tool, uses different extensions [12], as well as discrete event modelling tool Extend that also allows using extensions [2].

Communication library is an extension of simulation tool that enables to make connections with the adapter for sending and receiving data. Extensions can be made in different programming languages that depend on the features of simulation tool. For example, NetLogo extension was made using Java, but the extension for Extend is a DLL file, that was made in C programming language. The exchange of information between *communication library* and *communication adapter* is organised as simple as possible, as one can not predict in what programming language the extensions have to be made and what are the available features.

Communication Adapter is the element that communicates with models and *communication gateway,* and transmits them further to other communication adapters. Communication is carried out using a pre-defined data format, i.e. XML structured format, so that the data can be easily constructed by communication library. With communication adapter the user can define what information it will share with other members of communication network. Defining this information is a substantial step in communications process, as other users can observe what kind of information is available in within the network. When the user has defined this information to be shared, then communication adapters of other users will display a notification about newly available information that can be used in their models. For example, the user defines that he will share the data from the crop-harvesting model, and other users see in their communication adapters, that there is a possibility to receive data from this model.

Data storage is an element of communication adapter that keeps incoming data, transmitted through adapter. When the model is ready to use these data, then it demands them from communication adapter, this data that is being kept in data storage. *Data storage* can store also older data, if the model requires.

Communication gateway is the component that allows exchanging information between several computers. Without communication gateway information exchange is possible only within one computer. For the communication to be successful, the *communication gateway* has to be within a network infrastructure, so that communication adapters would be able to connect wit them. Communication adapters themselves do not have to necessarily be accessible from outside, because communication is performed through the gateway, not directly. *Communication gateway* maintains descriptive information about data available among models (not the data from models), it is registered in the communication gateway by the communication adapter. This adapter regularly gets into contact with the gateway and receives the up-to-date information about data available among models.

3.2 One Node Communication

Figure 1 shows a diagram of *communication node* that ensures communication at local level – within one computer. *Communication node* consists of *communication adapter* and models generated by simulation tool.

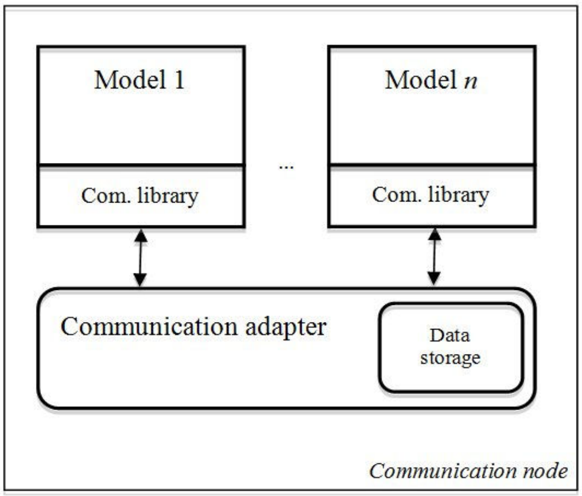

Fig. 1. Communication node

Between communication library and adapter within communication node two way communications is possible – receiving and transmission of data. Communication uses XML format, and that indicates also, which way of communication that is. If the model demands data, then the demanded data is returned, but this returned data is not XML anymore. Therefore it is easier for communication library to process the received data.

3.3 Multi Node Communications

Figure 2 shows the architecture for communication among several communication adapters in the framework of distributed simulation model, where nodes that can be located anywhere geographically. It is important that communication gateway can be seen by all adapters.

Also for communication between adapter and the gateway one uses XML messages. *Communication gateway* understands these kinds of action ("a") tags:

- Request for the list of available kinds of information (simulation data types, for example power of wind, direction of wind, etc);
- Registering of kinds of information;
- Subscription for simulation data;
- Sending of data.

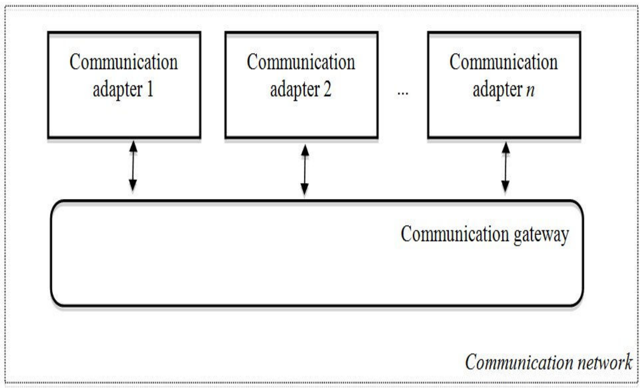

Fig. 2. Distributed communication network

Figure 3 shows the XML message for requesting the list of available kinds of information from simulations run within models. Adapter sends this kind of request frequently to *communication gateway*, so that the user can follow the up-to date information about available simulation results.

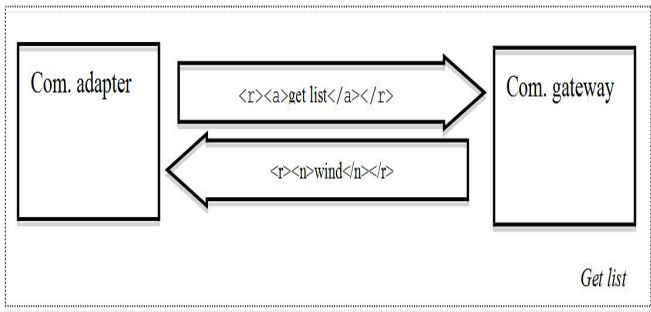

Fig. 3. Get list communication example

Figure 4 shows, how this information about available simulation information is being registered. Message has to indicate the type of action ("register"), as well as kind of information from the simulation ("wind"). Communication adapter after the registration receives a confirmation message, where "s" tag indicates status. If the status is "ok", then the registration was successful, if it shows "error", then no registration was performed. In this case "m" tag contains description of this error.

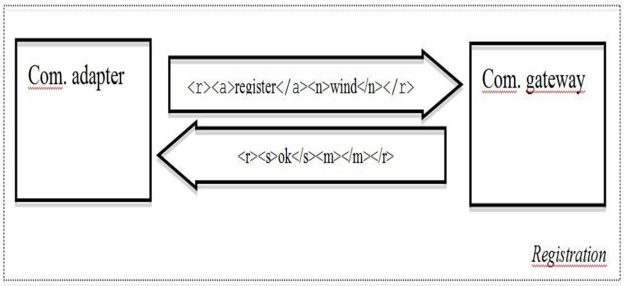

Fig. 4. Registration example

Figure 5 shows, how one subscribes for the simulation data. The example shows subscription for information about "wind". *Communication gateway* receives a confirmation message, where "s" tag indicates status. If the status is "ok", then the registration was successful, if it shows "error", then no registration was performed. In this case "m" tag contains description of this error.

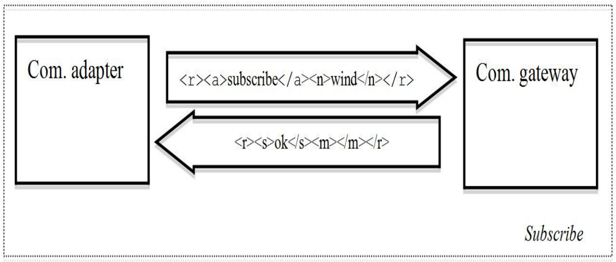

Fig. 5. Subscribe example

Figure 6 shows a situation, where communication adapter sends a message with type of action ("send"), kind of information ("wind") and value ("5"). When *communication gateway* receives the message with this information, it returns the status message to the adapter, and transmits the received information to all adapters that have subscribed for the certain simulation data ("wind" in this example).

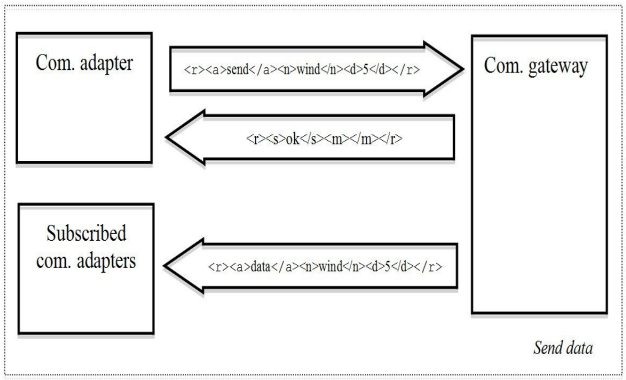

Fig. 6. Send/receive data example

Figure 7 illustrates the communication among adapters and activities of each adapter. Communication here is viewed from the side of adapters, without showing communication within the gateways.

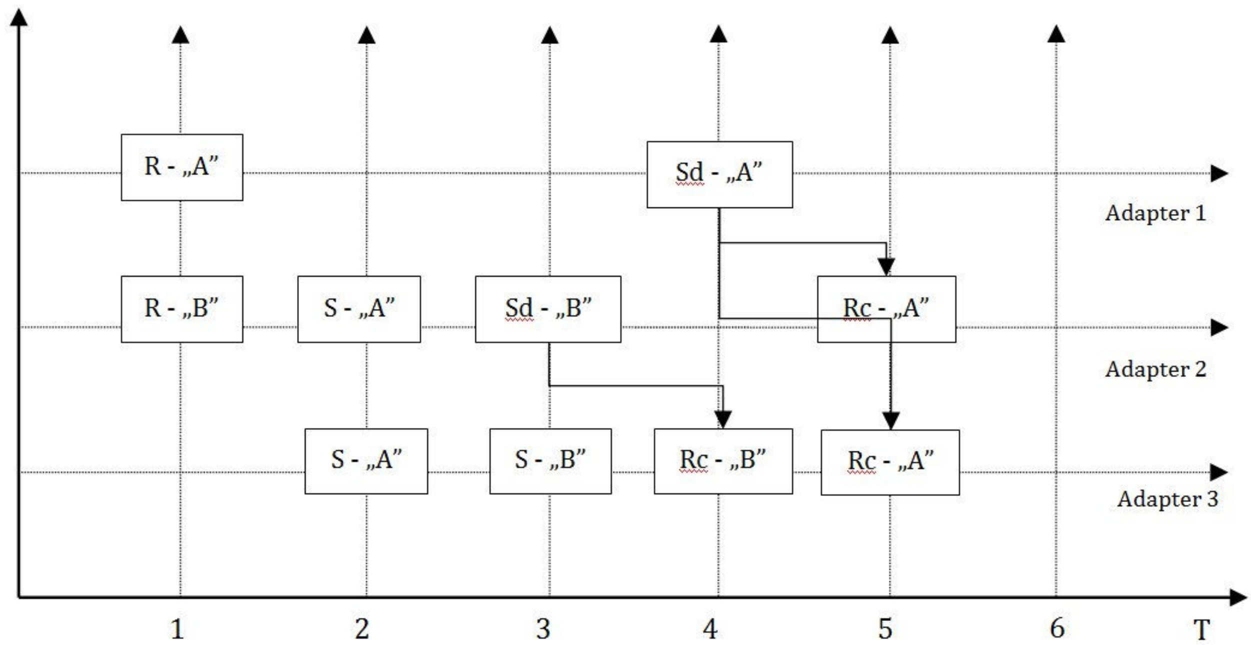

Fig. 7. Communications time diagram

The figure (see Fig. 7) shows communications system with three adapters. Adapter 1 performs only two activities in this example - registers interaction “A” and sends its data. Adapter 2 registers interaction “B” and subscribes for interaction “A” data. In the period of time 3 Adapter 2 sends data of interaction “B” and it is received by Adapter 3. Adapter 2 has subscribed for interaction “A”, and during period of time 5 it receives the data from Adapter1. During the period of time 2 Adapter 3 subscribes for interaction “A” and period of time 3 for interaction “B”. During the period of time 4 it receives data of interaction “B” that is being sent by Adapter 2, but during period of time 5 it receives data of interaction “A” that is sent by Adapter1.

4 Conclusion

Finally some important conclusions can be drawn regarding the distributed simulations and communication between models. First, it can be observed that the developers of simulation tools do not pay attention to the opportunities of models to communicate with other models, even of the same modelling environment. The paper outlines the structure of a comprehensible communication system and points out to its advantages. Second, the existent mechanisms do not offer time synchronization option; therefore there is a need to invent data storage. Further research should be aimed at searching for such time synchronising options. Third, future research should also seek for possibilities to ensure the transmission of more complicated data types. At the same time it should be noted that not all simulation tools offer the creation of extensions, therefore the comprehensible communication system described in this paper would not be applicable for such tools. Therefore if it is observed that simulation is revolutionizing the social and organizational sciences [11], then there are new challenges for engineering scientists in terms of enabling this process and empowering the researchers from other disciplines.

5 Acknowledgements

The current article was prepared in the framework of ERAF project No.2DP/2.1.1.2.0/10/APIA/VIAA /001 “Support for preparation of IST FP7 STREP project “Simulation Highway””.

References:

[1] A. Silins, E. Ginters, D. Aizstrauta, Easy communication environment for distributed simulation, Computational intellegence in business and economics, World scientific publishing Co. Pte. Ltd, 2010, 91-98.

[2] D. Krahl, The EXTEND Simulation Environment, in Proceedings of the 2002 Winter Simulation Conference/ E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.,(2002).

[3] E. Ginters, A. Silins, J. Andrusaitis, Communication in distributed simulation environment, In Proceedings of 6th WSEAS International Conference on Systems Science and Simulation in Engineering, Venice, Italy, (2007), 217-221.

[4] E. Ginters, A. Silins, Multi-level Approach for Environmental Systems Modelling in the Ligatne Natural Trails. WSEAS Transactions on Systems, Issue 4, Volume 6, (2007), 795-801.

[5] E. Ginters, A. Silins, Simulation Data Exchange in Distributed E-learning Environment, Proceedings of the 4th WSEAS/IASME International Conference on EDUCATIONAL TECHNOLOGIES (EDUTE'08), Corfu, Greece, (2008),138-143.

[6] H. Hibino, Y. Fukuda , A user support system for manufacturing system design using distributed simulation, Production Planning & Control, Vol. 17, No. 2,(2006), 128–142.

[7] H. Hibino, Y. Fukuda, Y. Yura, K. Mitsuyuki, K. Kaneda, Manufacturing adapter of distributed simulation systems using HLA, Simulation Conference, Proceedings of the Winter conf.,vol 2,(2002) 099-1107.

[8] HLA Blockset™, ForwardSim Inc, High Tech Park of Québec, http://www.forwardsim.com/en/prod_hlab.php, (2009).

[9] K. Pan, S.J. Turner, W. Cai, Z. Li, A Hybrid HLA Time Management Algorithm Based on Both Conditional and Unconditional Information, Parallel & Distrib. Comput. Center, Nanyang Technol. Univ., Singapore, (2008), 203-211.

[10] K. Perumalla, Tutorial. Handling Time Management Under the High Level Architecture, http://www.ornl.gov/~2ip/doc/perumalla-tutorialslides-iitsec06.pdf, (2009).

[11] K.M. Carley, Computational organizational science and organizational engineering, Simulation Modelling Practice and Theory 10 (2002) 253–269.

[12] U. Wilensky, NetLogo, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, http://ccl.northwestern.edu/netlogo/, (2009).