ROLES OF VISUALIZATION TOOLS IN DISTRIBUTED SIMULATIONS

Ing. Vladimír FLORIÁN, Doctoral Degree Programme (1)
Dept. of Intelligent Systems, FIT, VUT
E-mail: florian@fit.vutbr.cz

Ing. Pavel SLAVÍČEK, Doctoral Degree Programme (1)
Dept. of Intelligent Systems, FIT, VUT
E-mail: slavicek@fit.vutbr.cz

Supervised by: Dr. Zdeňka Rábová

ABSTRACT

This report deals with roles of visualization tools in distributed simulations. Describes advantages of using these tools in implementation phase and after implementation for modeling and simulations of complex system in distributed environment.

1 INTRODUCTION

Many problems are very often solved by simulations, these problems are usually complex and their simulations takes a lot of time. Internet become more accessible and gives a good bandwidth for communication in recent time, therefore distributed simulation is very interesting way for increasing speed of the simulations. Distributed simulation has many advantages than traditional simulation (speed up simulation, etc.), but on the other side has some disadvantages. Main disadvantage of the distributed simulation is complexity of simulation system. For this reason is important and very useful using some visualization tools already in implementation phase. Visualization tool can be used for monitoring actual state of distributed system, communications between submodels during simulation, analyse effectiveness of distributed simulation etc. Using of the visualization tools can help us to better understand negative factors for effectiveness of distributed simulation and helps us to minimize them.

2 DISTRIBUTED SIMULATION

Very interesting task is solving real problems. Tasks like this are often very complex and contain many non-linear feedbacks. It is very difficult to solve them with analytic methods and for this reason we are using theory of modeling and simulations for solving them. Traditional sequential simulations require a lot of computing power but not everyone has it. Solution of this problem is distributed simulation which can be executed in parallel on many personal computers connected via internet or local network and increase speed of the simulation. Distributed simulation is suitable solution how we can obtain a lot of cheap computing power. Today's, when internet become more accessible and gives a good bandwidth for communication, distributed simulation is very interesting way for simulations of complex systems.

In distributed simulation is necessary to decompose model into submodels with least dependencies. Each submodel has its own simulator and simulation can be done in parallel. We have to take into account this requirement of model decomposition when we choosing modeling and simulation techniques for our system. We had used DEVS formalism as a basic modeling and simulation technique for our distributed simulation system. Philosophy of the DEVS formalism consists in building components and connecting components of the model. This philosophy exactly correspond with distributed simulation requirement.

3 DEVS FORMALISM

In this section we present basic ideas and concepts of the DEVS formalism. Detailed information about this formalism can be found in [4][5].

We can describe models based on DEVS formalism in two levels. At the lower is model behavior described as sequence of deterministic transitions between model's states. This level is called atomic DEVS. At this level we describe elementary model's parts. Higher level is called coupled DEVS in this level is model described as a net of connected components. Components of this model can be either atomic DEVS model or coupled DEVS model. This feature enables building hierarchical and complex models or building libraries of commonly used submodels. Interconnection between two components represents their interrelations. Communication in higher level is based on the messages.

PARALLEL-DEVS

Parallel-DEVS differs from classical DEVS mainly in feasibility of all components activation, whose activation was planned at the same time, concurrently. This feature is essential in using models based on parallel DEVS in distributed simulation. The next mentioned useful feature is component architecture of the models. Parallel version of DEVS formalism corrects classical DEVS definition in both levels of model description. At the atomic level is added transition function called confluent, which decides about the next model state when a collision between internal and external event occurred. At the higher level is omitted function for component selection.

4 CONCEPT OF VISUALIZATION TOOL

Graphical visualization can be useful for some activities bind together with execution of parallel computing, such as analysis, verification, performance analysis of distributed simulation, development of new communication algorithm and protocols and models verification and debugging simulation. When are this information provided in textual form, is harder to understand them for users. Visualization systems can operate in different modes:

- on-line
- off-line
- combination of both previous

In online mode visualization system gains data from monitored simulation system in runtime and immediately presents the information. In offline mode the visualization system collets the data and stores them. After the simulation finished the stored data can be presented and analysed. Online mode has the advantage to show the real time execution of the monitored system, while offline mode allows post- processing of the collected data, which can be useful when the data are more complex and require more time to analyze. Developed visualization system for monitoring and analyzing of parallel simulation system based on DEVS should provide online as well as offline mode. Further this system must minimize the amount of computation perturbation to not to be the performance bottleneck of simulation system. Because trough the distributed simulation are solved complex problems, which time of simulation is long, this tool must be robust to make the monitoring of system possible for the whole simulation run.

When we need visualize this kinds of the data, is not suitable use one of existing visualization system for general purpose or conventional monitoring systems, because they could not satisfy all requirements specified for given system for distributed simulation. Better is to develop this system with use of suitable visualization library. Then this system will be better satisfy the requirements of analysis and adapts to changes of the requirements. Support for adaptability is important since requirements of users and enviromental changes force developers to constantly maintain and enhance their software. Visualization system should support following types of views:

- Environment view providing information of network computing enviroment, where the simulation is running. Information about used workstations, their loads, memory used for individual tasks.
- Information for evaluate performance and to help locate bottlenecks in system (for example how much time individual workstation use for computing, how much for message sending and waiting for reply).
- Debugging information, providing views into system activities to help locate bugs in program (message passing behavior between tasks).
- Interaction view providing support for evaluate results and for interaction with system.

For the best understanding of information must this visualization system provides several views simultaneously or the user can choose, which systems wants to monitor, if is several of them running concurrently, or which views on system wants to observe. This views will be updated in periodical intervals, the update time can user modify (this interval could not be too small to visualization system does not excessively perturb the monitoring system). Furthermore at the development of this system we must take respect to minimize ambiguity of providing information. This could be dangerous in some optimistic communication algorithms, which make possible out-of-order executions. Then the effectivity appears to be high, but the real matter is, that part of computing capacity is used for return to some of the previous consistent states in the case of algorithm failure.

4.1 ARCHITECTURE OF VISUALIZATION TOOL

From the aspect of flexibility at the development and the use of visualizaton system appears suitable the use of three-layer architecture. At the bottom layer are individual processes or threads running on each computing node used in distributed simulation, which will collect information from this computing nodes. Information about messages sends in this framework will be obtained by overloading methods for message passing in used communication interface. These data would be send on request or in periodic intervals to middle layer of visualization system, so called visualization server, that would store these data for post-processing and simulatenously provide them to user. Visualization server should provide support for filter incoming data to reduction and correlation. User interface, the last layer of system, will be called the visualization clients. These will provide displaying data to the user, possibility to set which data displayed and controlling of visualization server. This conception considerably ease development of system and his possible expand and modification. User then can choose from systems and factors which he wants to monitor.

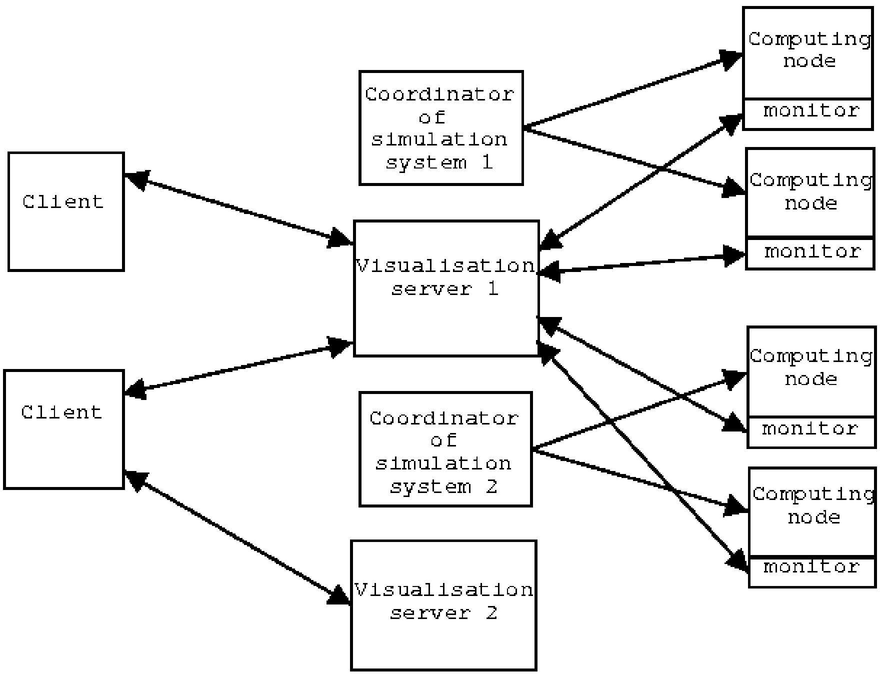

Figure 1: Architecture of visualization tool

5 CONCLUSION

This tool makes possible better view on the execution of distributed simulation, whereby ease development of system and later provides to user better understanding of system functions and help with debugging model. Obtained knowledge we put to use on implementation of visualization system, which will be helpful in developed distributed simulation system. As was mentioned at the introduction of this paper at the development process will be this tool used for debugging simulation system. Next will be this tool used along creation and debugging complex models. On construction of this framework we are preparing grant. In frame of this grant we will among other investigate possibilities of visualization in connection with distributed simulation and heterogeneity of models.

REFERENCES

[1] Vangheluve, H. : *Parallel-DEVS, a parallel, hierarchical, modular modeling formalism*, pages 716 to 722, Society of Computer Simulation International 1994, ISBN 0-7803-2109-X

[2] Carothers, D. : *Visualizing parallel simulations in network computing enviroments: a case study* Proceedings of the 1997 Winter Simulation Conference, 1997

[3] Kircher, M., Schmidt, C. : *DOVE a distributed object visualization eviroment* , C++ Report, March 1999

[4] Zeigler, P. :*Theory of Modeling and Simulations, second edition* , Academic press 2000, ISBN 0-12-778455-1

[5] Vangheluve, H. : *The Discrete EVent System specification (DEVS) formalism*, COMP 522 Fall Tem, 2002