An Efficient Statistical Synchronization Method For Parallel Simulation

Gábor Lencse

Department of Telecommunications
Széchenyi István University
Győr, Hungary
lencse@sze.hu

In this paper, we propose the improved Statistical Synchronization Method (SSM-T) for parallel discrete event simulation. Criteria are given for the time-driven approach (SSM-T). It is proven, that the level of the output error can be guaranteed. SSM-T is implemented in the OMNeT++ discrete event simulation tool, which is a useful and widespread framework for creating various simulation models to evaluate the performance of telecommunication networks. Case studies have been performed, which shows that SSM-T is a very efficient synchronization method for the parallel simulation of communication networks.

parallel discrete event simulation, statistical synchronization, applicability criteria, efficiency, accuracy

I. INTRODUCTION

Discrete event simulation is a powerful method in the performance analysis of communication networks, digital circuits and computer systems. The simulation of large and complex systems requires a large amount of memory and computing power that is often available only on a supercomputer. Efforts were made to use clusters of workstations or multiprocessor systems instead of supercomputers, as this would be much more cost effective. The simulation of large and complex networks is often a practical need when they are designed or analyzed. In many cases, the only option for the execution of the simulation is the use of a cluster of workstations. Due to the nature of the algorithm of the event driven discrete event simulation the parallelization is not an easy task.

The conventional synchronization methods for parallel discrete event simulation (e.g., conservative, optimistic) [2] use event-by-event synchronization and they are unfortunately not applicable to all cases, or do not provide the desirable speedup. The conservative method is efficient only if certain strict conditions are met. The most popular optimistic method "Time Warp" [3] often produces excessive rollbacks and inter-processor communication.

The Statistical Synchronization Method (SSM) [16] is a promising alternative to the conventional synchronization methods for parallel discrete event simulation. Unlike the conventional synchronization methods, SSM does not exchange individual messages between the segments but rather the statistical characteristics of the message flow. Actual messages are regenerated from the statistics at the receiving side. (Further explanation will be given later.) SSM claims to be less sensitive to communication delay and it requires less network bandwidth than event-by-event methods. Nevertheless, it is not accurate in the sense that an event that occurred in one segment of the system does not have an immediate influence on another segment. For this reason, the method cannot be applied in some simulations, for example in the case of digital circuits but remains feasible in other classes of simulation such as the performance estimation of the next generation networks.

An addition advantage of SSM is that it is relatively easy to extend existing non-parallel simulation software for use with SSM, which is not necessarily true for other synchronization methods. In this paper, we propose the improved Statistical Synchronization Method (SSM-T) for parallel discrete event simulation. SSM-T is implemented in the OMNeT++ discrete event simulation tool [8,10,18,19], which is a useful and widespread framework for creating various simulation models to evaluate the performance of telecommunication networks.

The remainder of this paper is organized as follows: after a brief introduction to SSM and SSM-T in Section II, the applicability criteria for SSM-T are given in an informal way in Section III. They are formalized in Section IV together with a proof that the required level of the output error can be guaranteed by satisfying the criteria. Next in Section V, we show positive and negative examples to give a better insight of the criteria. Afterwards in Section VI, we present the conditions for a good speed-up. Finally in section VII, we give the conclusions of the paper.

II. THE STATISTICAL SYNCHRONIZATION METHOD

A. The Original SSM

For those not familiar with SSM, a short summary of the Statistical Synchronization Method [16] is given here.

Similarly to other parallel discrete event simulation methods, the model to be simulated — which is more or less a precise representation of a real system — is divided into segments, where the segments usually describe the behavior of functional units of the real system. The communication of the segments can be represented by sending and receiving various messages. For SSM, each segment is equipped with one or more input and output interfaces. The messages generated in a given segment and to be processed in a different segment are

not transmitted there but the *output interfaces* (OIF) collect statistical data of them. The *input interfaces* (IIF) generate messages for the segments according to the statistical characteristics of the messages collected by the proper output interfaces (Fig. 1).
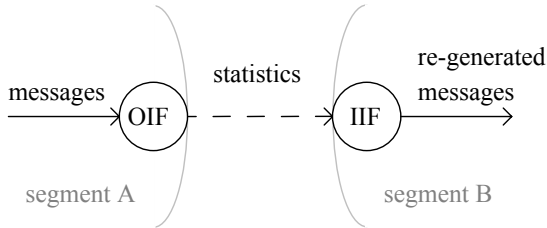

Figure 1.   An OIF - IIF pair

The segments with their input and output interfaces can be simulated separately on separate processors, giving statistically correct results. The events in one segment have not the same effect in other segments as in the original model, so the results collected during SSM are not exact. The precision depends on the partitioning of the model, on the accuracy of statistics collection and regeneration, and on the frequency of the statistics exchange among the processors.

B.   SSM-T: Refinement of SSM

The original SSM does not explicitly state when the OIF's should send their statistics to the appropriate IIF's. The results of [6,7] would suggest that the statistics collection must be continued until the sample contains the required number of observations, then the statistics should be sent and the statistics collection should be restarted. This was called *SSM-C* (the counter driven approach) in [5]. In that paper, *SSM-T* (the time driven approach) was proposed for parallel simulation, which works as follows. Using the notations of Fig. 1, at the beginning of the simulation the OIF of segment A must tell the IIF of segment B at what virtual time it will send the first statistics. This is t_1 in Fig. 2. In this figure the thin horizontal lines show the wall-clock (real) time of the processors executing the segments and the thick lines are the virtual times of the segments. Segment B takes into consideration the first statistics from segment A at t_1 virtual time. It is done in the following way: In the figure, segment B receives the first statistics from segment A at t_x (according to its own virtual time) and as $t_x < t_1$, segment B schedules the arrival of the statistics for t_1. The other possibility is shown on the example of t_2. Segment B has not received the statistics until t_2, and it has no more events scheduled with less than or equal time stamp, so the execution of the simulation of segment B is suspended until the statistics arrive from segment A. Then segment B receives the statistics and the execution resumes. Segment B always knows at what virtual time to expect the next statistics, because Segment A always includes the virtual arrival (=sending) time of the i-th statistics in the (i-1)th statistics package. We called this solution *loose synchronisation [5]*. This method makes it possible for the simulation of the segments to run independently on separate processors in the vast majority of time and therefore it may result in excellent speed-up.
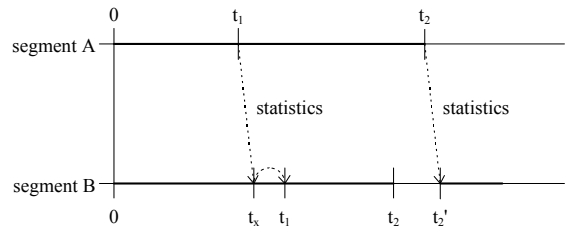

Figure 2.   The operation of SSM-T. See the text for explanation.

III.   THE APPLICATBILITY CRITERIA OF SSM-T

SSM-T can be applied and produces meaningful results if the following conditions are met:

(a) The simulated system can be divided into segments so that not the individual messages but only their statistical characteristics are important between the segments.

(b) A *small error* in the approximation of the statistical characteristics of the message flow causes *small error* in the output of the simulation that *depends only on the measure of the approximation error*.

(c) The parameters of the model may change during the simulation but the changes in the statistical characteristics of the message flows between the segments are *rare enough*.

(d) Note that a change in the statistical characteristics of the message flow is only propagated to other segments when the statistics package is actually sent over by the OIF. The fourth condition is that this delay causes error in the output of the simulation only during the delay and/or at most during an additional time interval with a length proportional to the delay.

IV.   THE OUTPUT ERROR OF SIMULATION WITH SSM-T

Let us denote the concerned statistical characteristics of the message flow by the random variable X and its approximation by X^*. The error of the approximation is also a random variable: $h_X = X^* - X$. The observed output of the simulation is denoted by O. The h_X error of the approximation causes $h_O = O^* - O$ error in the output. Condition (b) is defined formally as follows:

(b') $h_O = f(h_X)$ for some f, and $\forall \varepsilon > 0 \, \exists \delta: |h_X| < \delta \Rightarrow |h_O| < \varepsilon$.

Note, that sometimes the following conditions may be enough:

(b") $h_O = f(h_X)$ for some f, and $\forall \varepsilon > 0 \, \exists \delta: E\{h_X\} < \delta \Rightarrow E\{h_O\} < \varepsilon$, where $E\{\}$ is the expected value of the random variable.

Let us denote the number of observations in a sample by *n*. $\forall \delta > 0 \, \exists N: n > N \Rightarrow |h_X| < \delta$. The value of N depends on both the required value of δ and the convergence speed of the statistics collection method used. See [6] for more details about the convergence speed of some well-know statistics collection methods.

In the stationary case we can guarantee $|h_O| < \varepsilon$ with the appropriate choice of N.

Let us consider the transients in the system. If the distribution of the random variable X changes in segment A at t_c in the i-th sample collection interval, the exact new statistics arrive at segment B at the t_{i+1} synchronization point of virtual time (Fig. 3). If T_N is an upper bound for the length of any $(t_{i+1}-t_i)$ interval, then the length of the transient $T_{TR}<2T_N$.
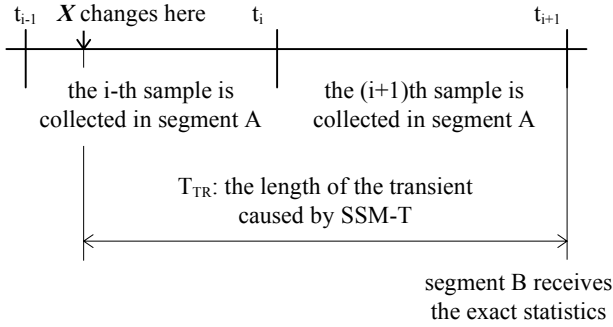

Figure 3.   The transient caused by SSM-T

Condition (d) says that the transient may cause output error during the T_{TR} time of the transient, plus maximum $c_e T_{TR}$ time after it. (c_e is an appropriate constant.) Thus the output of the simulation may contain an error due to the transient caused by SSM-T less than $c_e'T_N$ time, where $c_e'=2(1+c_e)$.

Let us denote the time elapses from the end of the $c_e'T_N$ time interval to the next change of X by T_{QST}, the time of the quasi stationary state. The empirical meaning of "*rare enough*" in condition (b) is: $c_e'T_N << T_{QST}$. Let us make it a bit more formal: H_O is an upper bound for the absolute value of the output error during the transient caused by SSM-T. Now, we show that $|\bar{h}_o|<\varepsilon$ can be ensured. Let us choose δ': $|h_X|<\delta' \Rightarrow$ $|h_o|<\frac{1}{2}\varepsilon$ during the T_{QST} time period. Let N_{TR} and N_{QST} denote the number of the collected output statistics values during the $c_e'T_N$ and the T_{QST} time periods, respectively. The before mentioned "*rare enough*" condition is:

(c') $N_{QST} \geq \dfrac{2N_{TR}H_O}{\varepsilon} - N_{TR}$

The average output error is:

$$\left|\bar{h}_o\right| \leq \left|\frac{N_{TR}H_O}{N_{QST}+N_{TR}}\right| + \left|\frac{N_{QST}h_o}{N_{QST}+N_{TR}}\right| \leq \tfrac{1}{2}\varepsilon + \tfrac{1}{2}\varepsilon \leq \varepsilon$$

However, the condition for N_{QST} is quite strict, and it is not always necessary. We have eliminated the error of the output of the simulation by averaging very many values. If we know the time of the changes of X in advance, or if we can detect the change, we can just omit the N_{TR} number of output statistics values with error. By doing so, a lot of virtual time (and therefore simulation execution time too) may be saved, because in this way only the $c_e'T_N$ virtual time is wasted, and the requirement for N_{QST} is just the same as it is in the case of the traditional event-by event synchronization:

(c") N_{QST} must be large enough to collect the statistics of the output of the simulation with the required accuracy.

The proportion of the wasted $c_e'T_N$ and useful T_{QST} virtual time is very important. The $c_e'T_N$ virtual time is used up just to eliminate the transient caused by SSM-T. In addition to its execution time comes the execution time wasted due to communication overhead between the processors executing the segments. However, if both of them are low compared to the execution time of T_{QST} and the model of the simulated system was partitioned in the way that the load of the executing processors is nearly balanced, our simulation may produce a good speed-up.

V.   EXAMPLES FOR THE APPLICATION OF SSM-T

A.   Satellite Power Consumption - A Positive Example

Let us consider the following example. A hurricane forecasting satellite collects data of the atmosphere and after some preprocessing, it sends them to the Earth for evaluation. The whole system is built up by three major functional units:

1. The Intelligent Measurement System controls the sensors and evaluates their signals, collects and preprocesses measurement data. Its output is a variable rate packet data flow. The packet rate depends on environmental conditions such as the state of the atmosphere, hurricane suspicious observations, etc.

2. The Data Transmission System carries the data packets from the Intelligent Measurement System to the Data Evaluation System on the Earth. The Data Transmission System contains a radio link downwards and another one upwards. The data sent through the downlink is acknowledged on the uplink. The transmission power is controlled in the function of the packet loss ratio, so the power consumption/bit depends on the environmental conditions (state of the atmosphere, orbit deviations, etc.) too. To save power, the carrier of the downlink is turned off when there is no transmission.

3. The Earth Data Evaluation System is responsible for the final evaluation of the collected measurement data.

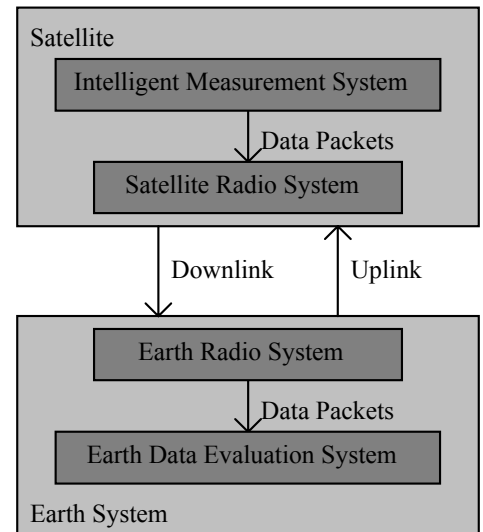Fig. 4 shows the block diagram of the physical system.

Figure 4.   The satellite data collection system

Our task is to determine the behavior of the power consumption of the Satellite Radio System to be able to tell the solar cell and battery requirements. The power consumption depends on both the packet rate from the Intelligent Measurement System and the radio channel conditions. Both quantities depend on environmental conditions that are too complex for an analytical treatment. Some of the environmental conditions (e.g. atmosphere) influence both the packet rate and the required transmission power, so they cannot be simulated independently. However, it is known from earlier experiments that the environmental conditions change very slowly compared to the data packet rate, that is typically millions of data packets are transmitted between two consecutive significant changes in the environmental conditions. It is also known that the channel capacity is more than twice as much than it is necessary for the maximum packet rate, so there is practically not buffering except that the packets are stored (to be able to retransmit them) until an acknowledgement is received. We propose the following parallel simulation:

- The Earth Data Evaluation System is omitted as it has no influence for the investigated power consumption.

- The remainder of the system is divided into two segments: the Intelligent Measurement System and the Data Transmission System, they are simulated parallel on two processors. The segments model those environmental conditions that are relevant to their operation.

- The two segments of the model are executed by two processors. SSM-T is applied between the two segments. The packet inter-arrival time statistics are collected by the OIF of the Intelligent Measurement System and the result is sent to the IIF of the Data Transmission System. The packet data flow is regenerated by the IIF of the Data Transmission System.

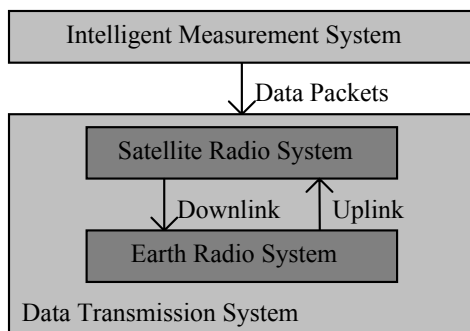Fig. 5 shows the block diagram of the simulation model.

Figure 5. The simulation model of the satellite data collection system

The applicability criteria of the SSM-T are satisfied:

(a) At the boundaries of the two segments, not the individual packets but only the average packet rate is important in the point of view of the power consumption.

(b) A small error of the estimation of the packet rate (or the distribution of the packet inter-arrival time) causes a small error in the calculation of the power consumption and depends only on the measure of the error, not the actual value of the packet rate.

(c) Significant changes are rare enough to make the necessary number of observations in quasi stationary state.

(d) As there are practically no buffering, the delay of the changes of the packet rate causes error only until the arrival of the new perfect statistics.

B. Negative Examples

The first example is the simulation of an FDDI [1] ring. If the ring is divided into two (or more) segments and the segments are simulated by separate processors, the explicit passing of the token cannot be replaced by the arrival time statistics of the token collected at the segment boundaries. This would be a violation of the Media Access Control protocol resulting in ring recovery (token loss, duplicate tokens etc.). Here SSM-T cannot be applied, because condition (a) is not satisfied.

Our second negative example is derived from the before mentioned positive one. Let us modify the system described there in the following way: The Data Transmission System contains two downlinks with no transmission power control. The unacknowledged packets are retransmitted until an acknowledgment arrives for the packet. The primary downlink is always operational and the secondary works only if it is necessary due to high packet rate and poor channel conditions (that is, a high number of packet retransmissions). The earth station does not have burst demodulators, so when the second carrier is put into operation, there is a significant synchronization overhead. For this reason, the carrier is not switched off immediately after transmitting a packet, but only after a certain delay. This also means that even a few packets on the secondary link may result in significantly increased power consumption. For any given channel conditions one can calculate what packet rate can completely exhaust the capacity of the primary downlink. An arbitrarily small error in the estimation of the packet rate from the Intelligent Measurement System can cause a serious error in the output of the simulation if the packet rate is close enough to the calculated critical rate. Now, condition (b) is not satisfied.

C. Two Interconnected FDDI Rings - Another Positive Example

In [5] SSM-T was used in the simulation of the FDDI backbone of the Technical University of Budapest. This network consists of two rings: The Northern Ring is a university-wide network and consists of 15 FDDI stations interconnected by 5 wiring concentrators. The Southern Ring is the backbone of the Faculty of Electrical Engineering and Informatics, and being smaller ring of 7 FDDI stations. The two rings are interconnected by a router. The topology of the network and the cable lengths were taken from the real system. The load used in the simulation model came from measurements taken on the real FDDI rings.

First, a very accurate simulation of the network was performed. Then, SSM-T interfaces were inserted between the

two FDDI rings and the simulation was performed with the same parameters as before. The utilization of the rings were measured in both cases. It was found that the utilization values were close to each other in the two simulations.

More detailed discussion of the experiments can be found in [5].

VI. Conditions for a Good Speed-up

In the practical application of SSM-T, the length of the virtual time interval while the OIF collects a statistics package is very important. There are three points:

1. The time interval should be large enough to collect a sample that is based on enough observations to produce an estimation with the required accuracy.

2. The time interval should be short enough to bound the length of the transient caused by SSM-T.

3. The frequency of the statistics exchange should not be too high to produce good speed up.

The first two conditions should be evident for the reader, but the third one requires some explanation. Until now, the simulation with SSM-T was independent from the execution environment. However, the main aim of SSM-T is to produce both good results and good speed-up. Thus, the overhead caused by the statistics exchange is very important. Let us consider this overhead:

The statistics transmission directed graph is defined as follows:

- the nodes are the segments of the simulation model,

- the (directed) edges are the segment to segment routes of statistics transmission.

If there are no directed loops in the graph then the simulation may work as a pipeline with infinite buffers between the stages. However, if there is a directed loop in the graph the virtual times of the segments of this loop are synchronized in some way. Let us consider the simplest example: there are two segments and they send statistics to each other. Let they exchange their statistics every T virtual time interval. The segments are executed by two processors A and B. The processors are identical and they do not have any other load. The execution time of the T virtual time is τ_A and τ_B. The time of communication is denoted by τ_C. This time includes the time of data packing and conversion to the network data format (e.g. XDR, if necessary) data transmission time and propagation delay plus data conversion from network format (if necessary) and unpacking. The execution time of a T virtual time interval with SSM-T is:

$$\tau_2 = \max(\tau_A, \tau_B) + \tau_C \tag{1}$$

The overhead of the statistics collection and regeneration done by the IIF's and OIF's is denoted by τ_{IIF} τ_{OIF}. These are included in τ_A and τ_B, so they must be substracted in the calculation of the execution time of the traditional simulation.

The execution time of a T long virtual time interval using traditional uni-processor simulation is:

$$\tau_1 = \tau_A - \tau_{OIF-A} - \tau_{IIF-A} + \tau_B - \tau_{OIF-A} - \tau_{IIF-A} + \tau_C \tag{2}$$

Let us group the I/O interfaces overhead into τ_{IF}.

$$\tau_{IF} = \tau_{OIF-A} + \tau_{IIF-A} + \tau_{OIF-A} + \tau_{IIF-A} \tag{3}$$

The speed-up is:

$$s = \frac{\tau_A + \tau_B - \tau_{IF}}{\max(\tau_A, \tau_B) + \tau_C} \tag{4}$$

This value can be close to 2 if $\tau_A \approx \tau_B$, $\tau_C \ll \tau_A$ and $\tau_{IF} \ll \tau_A$, that is the load of the processors is well balanced, the communication overhead, and the overhead caused by the statistics collection and message regeneration are small.

In a large (communication) system there are usually multiple points where the SSM-T applicability criteria are satisfied. It means that the insertion of the SSM-T interfaces to these points will not cause significant degradation of the accuracy of the results. Out of these points, the simulationist must carefully select those, that separate the model to segments of similar complexity, so that the computing power required by the simulation of the segments are in the same order.

When selecting the statistics collection method it is worth considering its algorithmic complexity [6]. An interesting new density estimation method, k-split [17] may also be used in SSM OIF's.

And last but not least the frequency of the statistics exchange should not be higher than it is required. In earlier experiments on simulating two interconnected FDDI rings by two processors using SSM-T [5], we achieved 1.75, 1.86 and 1.91 speed-up depending on the frequency of statistics exchange.

The existence of weaker criteria for the applicability of SSM-T is also a question of interest.

VII. Conclusion

The applicability of the modified Statistical Synchronization Method (SSM-T) was studied. Criteria were given for the applicability of SSM-T in parallel discrete event simulation. We have proven that the small level of output error of the parallel simulation using SSM-T compared to the uni-processor simulation without SSM-T can be guaranteed if the criteria are satisfied.

We showed a real life example where the applicability criteria are satisfied and SSM-T can be applied. We gave negative examples too. We presented the conditions when the application of SSM-T results in a good speed-up. The results confirm that SSM-T is a very efficient synchronization method in the parallel simulation of the communication networks.

One interesting step of the current work is to combine the Traffic Flow Analysis (TFA) [9] and the detailed event-by-event simulation [11] for the parallel execution of the combined system [12, 14, 15].

VIII. REFERENCES

[1] ANSI X3.139. Fiber Distributed Data Interface (FDDI) Token Ring Media Access Control (MAC). American National Standards Institute, New York, NY, 1987.

[2] R. M. Fujimoto, "Parallel Discrete Event Simulation." Communications of the ACM 33, no 10, 31-53, 1990.

[3] D. Jefferson et al., "Distributed Simulation and the Time Warp Operating System." Proceedings of the 12th SIGOPS - Symposium on Operating System Principles, 1987. pp. 73-93.

[4] G. Lencse, "Efficient Simulation of Large Systems - Transient Behaviour and Accuracy" Proceedings of the 1997 European Simulation Symposium (ESS'97) (Passau, Germany, Oct. 19-23, 1997.). SCS Europe, 660-665.

[5] G. Lencse, "Efficient Parallel Simulation with the Statistical Synchronization Method" Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation (CNDS'98) (San Diego, CA. Jan. 11-14, 1998.). SCS International, 3-8.

[6] G. Lencse, "Statistics Collection for the Statistical Synchronisation Method" Proceedings of the 1998 European Simulation Symposium (ESS'98) (Nottingham, UK. Oct. 26-28, 1998.). SCS Europe, 46-51.

[7] G. Lencse, "Applicability Criteria of the Statistical Synchronization Method" Proceedings of Communication Networks and Distributed Systems Conference (CNDS'99), (San Francisco, CA, USA, January 17-20, 1999.) SCS, 159-164

[8] G. Lencse, "Design Criterion for the Statistics Exchange Control Algorithm used in the Statistical Synchronization Method" Proceedings of the 32nd Annual Simulation Symposium (San Diego, CA, USA, April 11-15, 1999.) IEEE Computer Society, 138-144

[9] G. Lencse, "Traffic-Flow Analysis for Fast Performance Estimation of Communication Systems" Journal of Computing and Information Technology Vol. 9, No. 1, (2001.) 15-27.

[10] G. Lencse, "Parallel Simulation with OMNeT++ using the Statistical Synchronization Method" Proceedings of the 2nd International OMNeT++ Workshop (Technical University Berlin, Berlin, Germany, Jan. 8-9, 2002.) 24-32.

[11] G. Lencse, "Combination and Interworking of Traffic-Flow Analysis and Event-Driven Discrete Event Simulation" Proceedings of the 2004 European Simulation and Modelling Conference (ESM'2004) (Paris, France, Oct. 25-27, 2004.) EUROSIS-ETI, 89-93.

[12] G. Lencse, "Speeding up the Performance Analysis of Communication Systems" Proceedings of the 2005 European Simulation and Modelling Conference (ESM'2005) (Porto, Portugal, Oct. 24-26, 2005.) EUROSIS-ETI, 329-333.

[13] G. Lencse and L. Muka, "Convergence of the Key Algorithm of Traffic-Flow Analysis" Journal of Computing and Information Technology, Vol. 14, No. 2, (June 2006.) pp. 133-140.

[14] G. Lencse and L. Muka, "Combination and Interworking of Four Modelling Methods for Infocommunications and Business Process Systems" Proceedings of the 2007 Industrial Simulation Conference (ISC'2007) (Delft, The Netherlands, June 11-13, 2007.) EUROSIS-ETI, 350-354.

[15] L. Muka and G. Lencse, "Decision Support Method for Efficient Sequential and Parallel Simulation: Time Decomposition in Modified Conceptual Models" Proceedings of the 2007 European Simulation and Modelling Conference (ESM'2007) (Malta, Oct. 22-24, 2007.) EUROSIS-ETI, 574-581.

[16] Gy. Pongor, "Statistical Synchronization: a Different Approach of Parallel Discrete Event Simulation" Proceedings of the 1992 European Simulation Symposium (ESS 92) (The Blockhaus, Dresden, Germany, Nov. 5-8, 1992.) SCS Europe, 125-129.

[17] A. Varga, "K-split – On-Line Density Estimation for Simulation Result Collection". Proceedings of the European Simulation Symposium (Nottingham, UK. Oct. 26-28, 1998.) SCS Europe, 41-45.

[18] Y. A. Sekercioglu, A. Varga, and G.K. Egan, (2003), "Parallel simulation made easy with OMNeT++" Proceedings of the European Simulation Symposium (ESS'2003), (Delft, The Netherlands, Oct. 2003.) SCS, 2003.

[19] M. Kozlovszky, Á. Balaskó and A. Varga, "Enabling OMNeT++-based simulations on Grid Systems". Proceedings of the 2nd International Workshop on OMNeT++ (OMNeT++ 2009), (Rome, Italy, Mar. 2009.) ICST, 2009.

[20] http://www.omnetpp.org

[21] http://www.omnest.com

BIOGRAPHY

GÁBOR LENCSE received his M.Sc. in electrical engineering and computer systems at the Technical University of Budapest in 1994 and his Ph.D. in 2001. The area of his research is (parallel) discrete event simulation methodology. He is interested in the acceleration of the simulation of info-communication systems. Since 1997, he has been working for the Széchenyi István University in Győr. He teaches computer networks and networking protocols. Now, he is an Associate Professor. He is a founding member of the Multidisciplinary Doctoral School of Engineering, Modelling and Development of Infrastructural Systems at the Széchenyi István University. He does R&D in the field of the simulation of communication systems for the Elassys Consulting Ltd. since 1998. Dr Lencse has been working part time at the Budapest University of Technology and Economics (the former Technical University of Budapest) since 2005. There he teaches computer architectures.