A DEVS-BASED FRAMEWORK FOR SIMULATION OPTIMIZATION: CASE STUDY OF LINK-11 GATEWAY PARAMETER TUNING

Hojun Lee, Bernard P. Zeigler
Arizona Center of Integrative Modeling & Simulation
The University of Arizona
Tucson, AZ

and

Doohwan Kim
RTSync Corp.
Phoenix, AZ

ABSTRACT

Discrete Event System Specifications (DEVS) is a mathematical formalism based on system theoretic principles, which has evolved with state-of-technologies implementation over the past few decades. In this paper, we discuss a DEVS framework to solve parameter optimization problems. As a case study, we consider the Link-11 gateway that has been developed by the Joint Interoperability Test Command (JITC) is to provide interoperability between Link-11 network and TCP/IP network. The performance of Link-11 gateway is highly sensitive to the sampling rate of soundcards since the frame time of Link-11 signal is very short. Unfortunately, the sampling rate is not as accurate as it is specified by the manufacture of soundcards. A solution is to search for an optimized parameter that can be used to adjust the sampling rate. We apply an optimization technique to search for optimal sampling rate in modeling and simulation environment, DEVSJAVA. The DEVS-based framework can facilitate efficient global optimal parameter search capability and reduce execution time benefiting from its parallel and variable structure implementation.

INTRODUCTION

In many applications, it is difficult or impossible to express a system or its behaviors in analytical fashion. Simulation optimization is a process to explore a best value of some decision variables via simulation process for complex systems which are not easily formulated in analytic expressions [1][2].

Discrete Event System Specification (DEVS) is an advanced and well-defined mathematical modeling and simulation formalism based on system theory [3]. For decades, DEVS has been applied to diverse modeling and simulation problems with various extensions such as Dynamic Structure DEVS, Symbolic DEVS, Fuzzy DEVS, and Real-Time DEVS [3]. DEVSJAVA, which is a DEVS modeling and simulation environment in Java, supports the implementation of the various DEVS extended formalism [4].

In this paper, we propose a DEVS-based framework for simulation optimization and provide a proof-of-concept employing DEVSJAVA with an application of Link-11 gateway (GW) [5]. The performance of the gateway is mainly affected by a sampling rate specification of soundcards which is not as correct as it is expected to be. Since it is almost impossible to vary the specification in detail, we adjust a parameter relevant to the sampling rate. The experimental results show that the approaches can be applied to the given parameter tuning problem successfully.

We provide the motivation of the DEVS-based simulation optimization including overview of this gateway with basic fundamentals of Link-11 signal and design concept of the gateway in the following section. Then we show some details about DEVS modeling and experimental results of the proposed approach. An advanced approach that employs dynamic structure is also discussed. Finally, we conclude with discussion of future work employing distributed simulation via DEVS/SOA [6].

MOTIVATION OF SIMUATION OPTIMIZATION: PARAMETER TUNING FOR A SOUNDCARD

Overview of the Link-11 GW

The Link-11 is a variation of Tactical Digital Information Links (TADIL) series. It transmits binary data over RF network based on a digital modulation technique such as Quadrature Phase Shift Keying (DQPSK). This enables participants in the network to communicate through HF Radio equipment in the normal operational environment. To facilitate operation or testing, some different methods are devised to connect various players using Link-11 messaging (TADIL A) over analog wireline or digital link and satellite [7]. Interoperability of Link-11 is still an interesting issue in a variety of military communication systems. It is important for the test community to cost-effectively implement tactical data connectivity of this kind over widely employed TCP/IP data networks. We devise a gateway that allows us to connect the Link-11 Data Terminal Sets audio (analog) input and output through analog-to-digital conversion and decoding to such networks. The gateway was built to replace the RF

978-1-4244-2677-5/08/$25.00 ©2008 IEEE

equipment so that we can deliver Link-11 data over TCP/IP network as shown in Figure 1 and Figure 2.

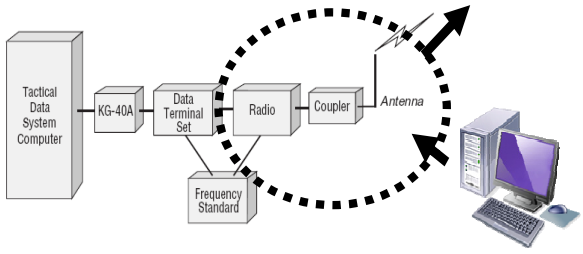


Figure 1. The replacement of RF network with the gateway

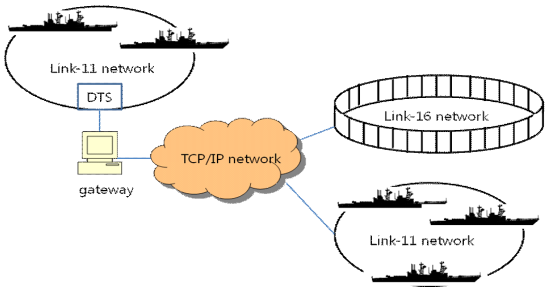


Figure 2. Overview diagram of the gateway concept

The gateway can enlarge network footprint and enhance connectivity to other networks including Link-11 and Link-16.

Design concept of building the gateway

In general, Link-11 transmits or receives data over either a RF network (HF Radio) or a voice channel (audio to and from the Data Terminal Set), and the corresponding terminal equipment to participate. We use the Link-11 DTS, as participants normally do, but digitize the audio for transmit into a digital network (TCP/IP), or conversely converts IP packets into the equivalent audio into the DTS. This enables distributed Link-11 players to participate over a TCP/IP network connection, which provides higher reliability than a dial-up audio line that has been traditionally used.

Actual implementation of this IP gateway is accomplished by decoding modulated audio signal through a PC-sound card, and packing and sending the bit stream over IP network. Conversely, the gateway receives packets from the IP network, which it encodes into audio and sends to the DTS. Thus the gateway is transmitter, receiver and client on the IP network as shown in Figure 3.

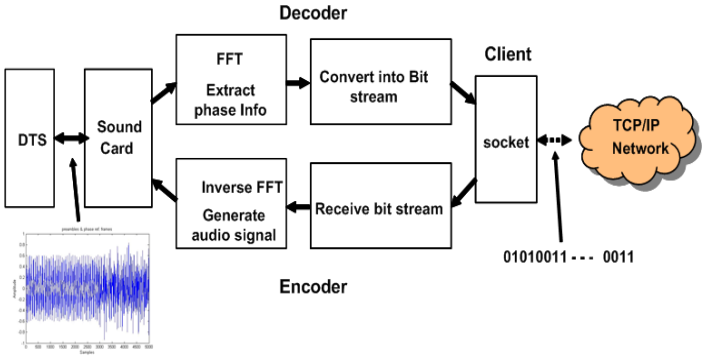


Figure 3. Design concept of Link-11GW

Encoding and decoding techniques for the gateway

Conventional encoding and decoding processes are based on matched filter or correlation techniques. An alternative is based on Digital Signal Processing (DSP) using a Fast Fourier Transform (FFT), to manipulate frequency components within the real-time audio stream. Since the specific frequencies that contain information are known, the FFT process can produce a set of complex numbers that correspond to frequency components. The complex numbers carry the information about the signal in terms of magnitude and phase, which can be used to extract phase information from the frequency components. With modern computing platforms, this technique can be processed in real time, so that we can handle every signal from DTS using a Discrete Fourier Transform [8].

For the encoding process, we use an inverse FFT that converts the phase change to audio signal. The audio signal is a real-time signal so we must accordingly place complex numbers in the bin of the FFT in a symmetric pattern. For the required magnitudes, an automatic gain control process is performed to ensure the power difference of 6 dB between tones. The FFT method greatly simplifies analysis and handling of the audio signals without any loss of information. The audio signal coming from DTS is sampled through the PC sound card at a 44,100 Hz sampling rate. I then process these digital discrete samples based on the method of DSP as described.

A challenge and its solution based on DEVS Experimental Frame (EF)

During the gateway testing, it was found that the sampling rate of a commercial soundcard did not match with its standard specification precisely. More specifically, the actual sampling rate of the soundcard that were used is not exactly same of the hardware specification given by manufacturer. For example, the gateway seemed to operate at the sampling rate of 43,998.86 Hz not 44,100 Hz on the test machine. The one frame time of Link-11 signal is so short that the performance of the gateway is very

susceptible to the accuracy of sampling rate of the soundcard. Unfortunately, adjustment of the sampling rate in finer detail level is not available for commercial soundcards since they come with a few number of fixed sampling rates such as 44,100, 48,000 and 96,000 Hz. With exhaustive trial and error methods, it's almost impossible to search for compensating (correcting) parameter. Phase shift is only difference between frames. We tried several methods to detect the shift and there was no applicable one. So we need an optimization technique for the problem. Furthermore due to absence of an analytic function for the parameter value this optimization problem requires a simulation-based approach. The DEVS simulation framework allows us to reuse the same model structures to obtain the compensation value each time for different soundcards.

In this problem, we need a parameter which can reflect the sampling rate's variation instead of adjusting the sampling rate itself since it can not be easily handled. The parameter is the time length of one frame. However, as we handle digital signal, in fact samples, the problem of figuring out frame time of the signal is equivalent to the problem of finding the number of samples of the signal after sampling process.

DEVS Formalism

The formalism for an atomic model and a coupled model is shown below [3]:

Atomic model:

$M = <X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta>$ (Equation. 1)

where,

X: a set of inputs;
S: a set of states;
Y: a set of outputs;
δ_{int}: Internal transition function;
δ_{ext} : External transition function;
λ: Output Function;
ta : Time advance function.

Coupled model:

$DN = < X, Y, D, \{M_i\}, \{I_i\}, \{Z_{i,j}\} >$ (Equation. 2)

where,

X: a set of external input events;
Y: a set of outputs;
D: a set of components names, for each i in D;
M_i: a component model;
I_i: the set of influences for I; for each j in I_i;
$Z_{i,j}$: the i-to-j output translation function.

DEVS provides an efficient simulation framework via the Experimental Frame (EF), in which we can define the model of a certain system configuration and run a simulation. This framework can also be plugged into the target system under testing i.e. the Link-11 Gateway.

To obtain exact number of samples, experimentations run with different parameter values. The Generator model (shown in Figure 4) generates input segments for gateway by importing the stored real audio data or generating the input segments of its own. It computes the signal data with the arbitrary number of samples per frames and feeds it to the gateway (processor). The Transducer model shown in Figure 4 has a stored reference bit pattern of NETTEST, to which it compares the output bit stream of the gateway. Controller model monitors the result of the bit-wise comparison and controls the whole simulation process. With many iterations of the EF models until the bit-wise comparison produces zero errors, we can obtain final optimal frame-sample value.

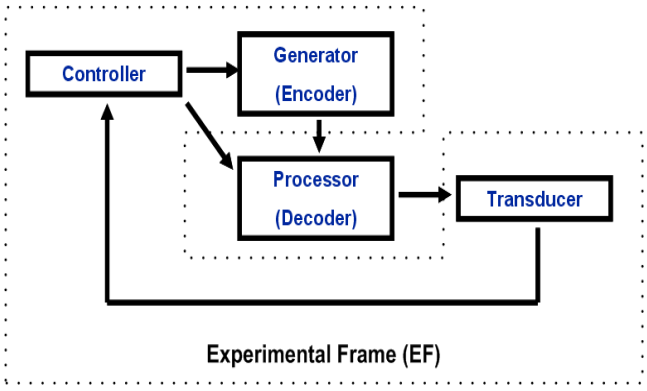


Figure 4. DEVS EF for the parameter optimization of the gateway

PROBLEM BACKGROUND

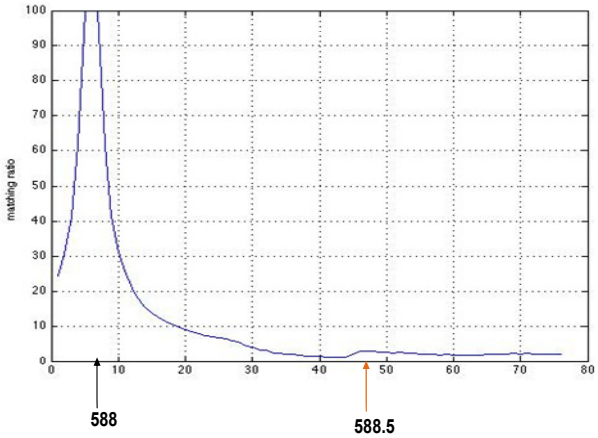


Figure 5. Search space for Link-11GW parameter tuning

Some simulation optimization techniques that can be applied to this type of problem are discussed in [1][2].

Since gradient information is available as shown in Figure 5, we use the discrete version of stochastic approximation algorithm that employs estimated gradient information computed with two different points, called a gradient-based procedure [1][2].

In Figure 5, the Y axis represents the matching ratio which is compared results computed by the transducer. The X axis represents the number of samples per frame. For two decimal points accuracy, the search space is discrete with values such as 588.00, 588.01, and 588.99

There are three optimization components we set up: *Decision variables, Objective function, and Constraints* [2]. The final outcome value of the simulation is the number of samples per frame, so the value is the decision variable. In this problem, we can not formulize an objective function so that we adopt simulation instead. Finally, we need to set up some constraints. Constraints determine the boundary of a search area. we restrict the searching range from 585.00 to 591.00 since 588.00 is the ideal value and we assume that ±3.00 is wide enough to compensate for bilateral variation of the decision variable. As shown in Figure 5 the gradient gets steep when it comes close to the optimal point. In addition, two points (587.99 and 588.01) near the optimal point (588.00) have the similar matching values. So we need to get step size smaller, as it approaches to the optimal point. The estimated gradient is calculated by two distinctive points. The difference of the two points is 0.01, since it gives good distinctive gradient in this case.

System Entity Structure for DEVS models architecture

To implement the experimental framework in DEVSJAVA, System Entity Structure (SES) is employed to represent hierarchical structure of models. Representing a family of hierarchical DEVS models, SES consists of elements and relationship that are represented by tree-type structure as shown in Figure 6 [3][9].

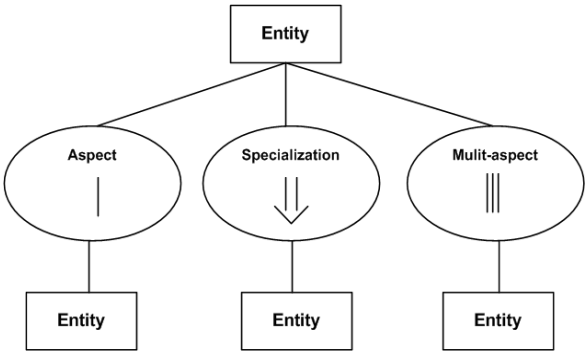


Figure 6. Basic SES representation

Entities represent things in the real or imagined world. Aspects represent ways to decompose things into sub-components. Multi-aspects are aspects for which the components are all of one kind. Specializations represent categories or families of specific forms that a thing can assume. The Link-11 Gateway experimental framework's architecture in SES is shown in Figure 7.

The generator in EF is implemented as a transmitter and the processor in EF is mapped to a receiver in Link-11GW. The controller plays a role of guiding simulation process for the search algorithm. The SES representation helps developers construct the models or components to be generated and their relationship in the hierarchical structure.

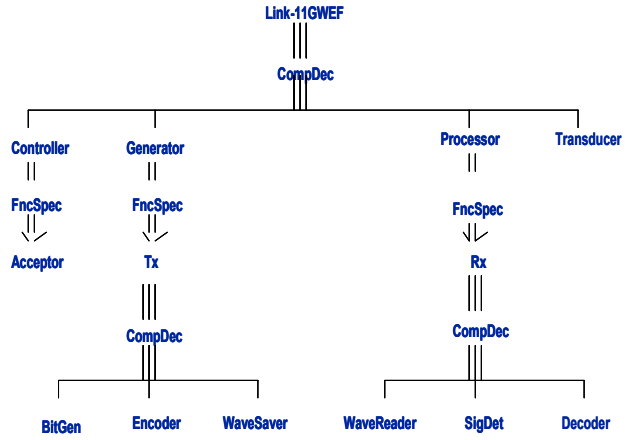


Figure 7. SES representation for Link-11GW EF

DEVS modeling in DEVSJAVA

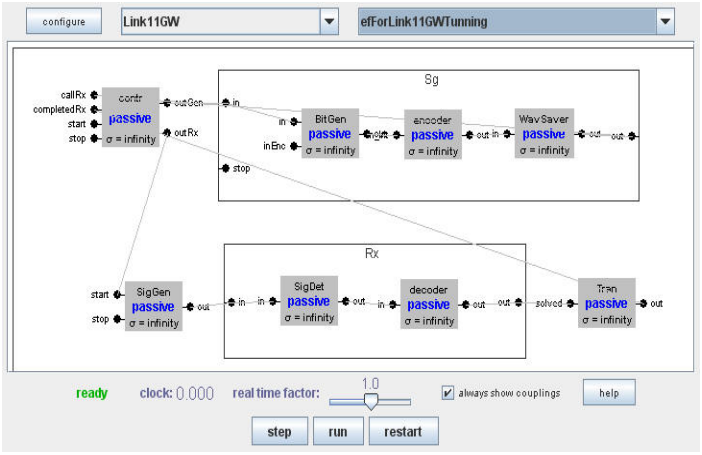


Figure 8. DEVS models of Link11GW EF in Simview

The EF is implemented in DEVSJAVA [4]. DEVSJAVA is a Java-based developing environment to realize DEVS

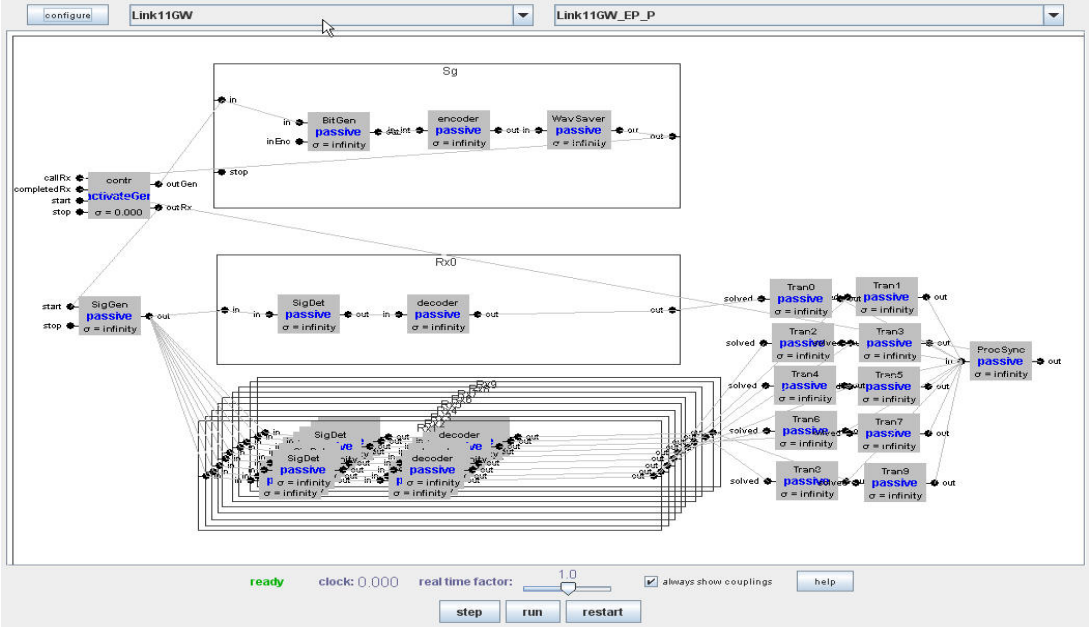


Figure 9. DEVS models for parallel search in Simview

formalism of (Equation 1 and Equation 2). It provides a simulation environment called Simview that visualizes the DEVS models and allows developers to verify the models and carry out simulation in graphical environment. It shows all models, its hierarchical structure and some information such as coupling. The viewer demonstrates models' behaviors such as state change and message exchange between models as well. The modeling of the Link-11GW EF in DEVSJAVA is shown in Figure 8.

Experiments 1

We tested the performance of the suggested experimental frame with the two difference source of input wave file. First, the generator (transmitter) generates Link-11 audio signal with an arbitrary parameter and stores the signal as a wave file. Then, the processor (receiver) loads the wave file and carries out decoding process. The transducer verifies the simulation results against the reference. The evaluation data are received by the controller that computes new test parameter and runs the simulation again until the best value is found. We generated input signal with two values: 588.00 and 588.02. The experimental frame results indicate the correct simulation with the two values. For the second approach, we conduct experiments with a real Link-11 wave file which was recorded directly from DTS. The result has shown that the given framework could also search the parameter of the recorded signal: 588.01.

After obtaining the parameter of the recorded signal we tested the result on the real gateway. We play the recorded signal with the windows media player on one machine to emulate DTS connected with the other machine that executes the gateway decoder. With the parameter previously found, 588.01, the decoder has shown the correct match.

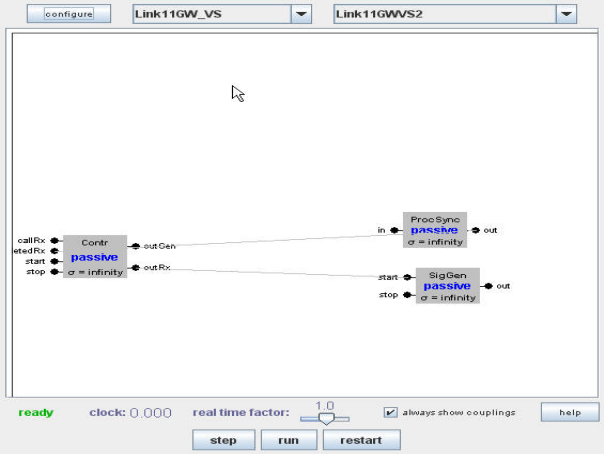
ANOTHER CHALLENGE: GLOBAL OPTIMUM VS. LOCAL OPTIMA

Basic approach: Parallel processing for divide and conquer

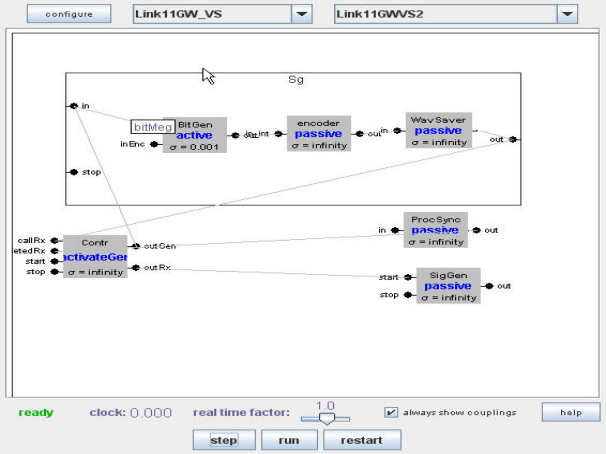
As in the Figure 5, it's important to start with a global optimal parameter number (global optimum, 588) rather than local optima (588.5) in the parameter search space.

Our strategy to avoid starting with local optimal parameter number is to divide the search space and examine the subregions in parallel processing paradigm: divide and conquer. Although being considered as an alternative, random search may take longer time to travel the search region until it finds a good start point. With the parallel search capability supported by DEVSJAVA, it is more efficient to test the whole region in shorter time.

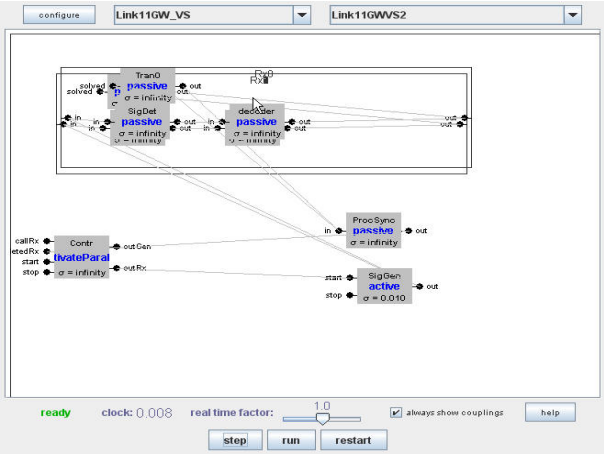
After several trials, we found that 0.6 is good distance between initial test values to avoid local optimum. If we break down the region by 0.6 we need 10 processors to cover the whole search space. Figure 9 shows DEVS models with 10 processors. There are two processing phases in finding global optimum: parallel search and optimization phase. The artificial signal generated by the transmitter goes through parallel search phase on each processor with different parameters. After parallel processing the controller chooses a decent parameter value



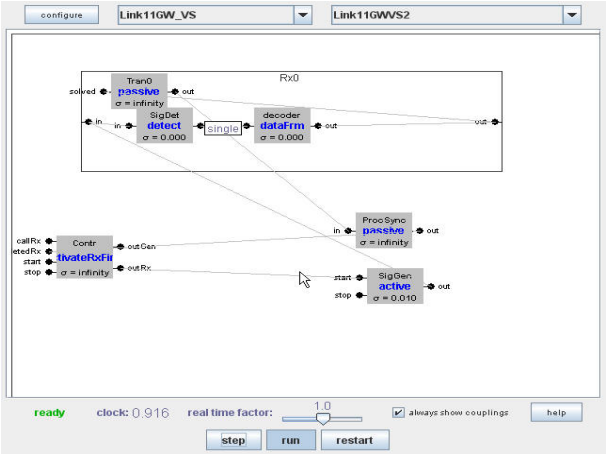
(a) basic components



(b) synthetic signal generation



(c) parallel search phase



(d) optimization phase

Figure 10. Variable Structure DEVS models.

based on the first simulation results. Then the simulation goes through optimization phase to discover the optimal value in single processing, which means only one processor is needed in this phase. There is one wavreader for 10 processors since every processor uses the same input data.

Advanced approach: Variable Structure Modeling

To be more efficient in terms of time consumption, we consider a different modeling approach. At each phase, we just need certain components. The others are not necessary. If we create components that are essential at certain time to carry out simulation process without unnecessary ones, we expect to reduce message traffic in DEVS simulation protocols. The computation burden is relieved as well.

This variable or dynamic structure concept is implemented in DEVSJAVA [10]. In variable structure, the components are created or deleted dynamically according to state changes. In addition, some modeling information is changed without creating or deleting components. Variable Structure DEVS supports the following operations: addModel(…), removeModel(…), addCoupling(…), removeCoupling(…), addInport(…), addOutport(…), removeInport(…), and removeOutport(…).

We modify the modeling structure and create the models in Figure 10. Before starting the whole process, there are three basic components (Figure 10. (a)). First, we need input data so only create signal generation components and make couplings (Figure 10. (b)). Then, for parallel search we bring 10 processors up after removing signal generation components (Figure 10. (c)). After getting a good initial start point, we carry out optimization phase with one processor as in Figure 10. (d).

Experiments 2

We generated three 160 second long signals with three test values: 586.65, 588.01, and 589.95. In the search phase, we set up a threshold at 2% of matching ratio in order to pick up one parameter value among multiple simulation

outputs. The target matching ratio in optimization phase is 100%. The controller changes the test parameter until it achieves the goal. Table 1 shows the initial computation time in finding the global optimum.

Table 1. Computation time for parallel and single process

Phase	Computation time (sec)
Search phase with 10 processors	199.194
optimization phase with single processor	43.987

We tested Variable Structure under the same experiment specifications. The following Table 2 shows that with variable structure we can reduce simulation time by 0.01% in parallel and 41% in single.

Table 2. Computation time of Variable Structure

Phase	Computation time (sec)
Search phase with 10 processors	197.289
optimization phase with single processor	26.04

The intention of variable structure is, in fact, to cut down the computation loads on multiple processors. The results, however, turns out the decrease of time in single processing. The reason is that the number of components in single process is much less than that in parallel process, which leads to less intensive message transferring between models in DEVS simulation protocols.

FUTURE WORK

We intend to implement parallel / distributed optimization simulations to achieve speedup of computation time using DEVS/SOA environment, an implementation of DEVS to provide web-based M&S services employing the infrastructure and standards of the Service Oriented Architecture. DEVS/SOA [6] supports the deployment of information-sharing DEVS Agents.

CONCLUSIONS

In this paper, a DEVS-based simulation optimization framework was discussed and implemented to find the parameter value for the Link-11 gateway to work properly. The experimental results show that DEVSJAVA is general and effective simulation optimization environment since it supports various extended DEVS formalisms including Variable Structure DEVS to carry out cost-effective simulation in time and resources.

We expect that further extension of this study to distributed simulation will bring more effective and faster simulation optimization.

REFERENCES

[1] Michael C. Fu, Fred W. Glover, Jay April, "SIMULATION OPTIMIZATION: A REVIEW, NEW DEVELOPMENTS, AND APPLICATIONS," Proceedings of the 2005 Winter Simulation Conference.

[2] S. Ólafsson, J. Kim, "SIMULATION OPTIMIZATION**,"** Proceedings of the 2002 Winter Simulation Conference.

[3] Zeigler, B.P., Kim, T.G., and Praehofer, H., Theory of Modeling and Simulation, 2nd ed., Academic Press, New York, 2000.

[4] Arizona Center for Integrative Modeling and Simulation, DEVSJAVA, (Accessed 2008, June, 11) [Online], Available: http://www.acims.arizona.edu/SOFTWARE/software.shtml#DEVSJAVA

[5] H.J. Lee, Taekyu Kim , Zeigler, B.P. , Dale Fulton, Doohwan Kim, "Improving Testing Capability of Interoperability for Link-11 by building a Gateway for a TCP/IP Network", SISO 2007 Fall Simulation Interoperability Workshop, Volume 2, pages 663-669, Orlando, FL, September 2007.

[6] Mittal, S., Risco-Martin, J.L., and Zeigler, B.P., "DEVS-Based Simulation Web Services for Net-centric T&E", Summer Computer Simulation Conference SCSC'07, July 2007.

[7] Navy Center for Tactical Systems Interoperability (NCTSI): "Understanding Link-11: A Guidebook for Operators, Technicians, and Net Managers", pp. 2_1- 2_31, September 1996.

[8] John G. Proakis, Dimitris Manolakis: Digital Signal Processing:Principles, Algorithms and Applications , 3rd ed., Prentice Hall, 2005.

[9] Zeigler, B.P. and Hammonds, P.E., Modeling & Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange, Elsevier, 2007.

[10] Xiaolin Hu, Bernard P. Zeigler, Saurabh Mittal, "Variable Structure in DEVS Components-Based Modeling and Simulation, " SIMULATION, Vol. 81, Issue 2, Feb. 2005, pp. 91-102.