

MATLAB Simevent: A Process Model Approach for Event-Based Communication Network Design, (A Case for Reengineered DCN)

Prof. H. C Inyama¹, Udeze Chidiebele. C², Dr C.C Okezie³, Okafor Kennedy. C.⁴

^{1 & 3} Department of Electronic and Computer Engineering, Nnamdi Azikiwe University Awka Nigeria
^{2 & 4} Electronics Development Institute (FMST-NASENI) Awka Nigeria

ABSTRACT

Contemporarily, the practice of science and engineering in relation to complex event based system design and implementation calls for a method of accurately verifying, modifying and deploying such a system to enable analysts predict the effect of changes to the system stability. Modeling is commonly used for understanding engineering and operations of systems in general. Developing a system whose operation is represented as a chronological sequence of events is the basis of Event based Engineering (EBE). The complexity associated with the design of such systems can be reduced through modeling, visualization and Discrete Event Simulation (DES). Lift systems, Queue models, Flow control models, communication networks etc, can accurately be developed and analyzed using DES. This paper presents modeling with event based simulation as a simplified approach for developing complex event-based engineering systems. It outlines steps to developing a simulation model as well as its analysis. A case for a communication process model (Reengineered datacenter network- DCN) is presented using MATLAB Simevent software package. The discussion presented in this paper will help engineers, system architects, computer scientists to understand the basis of modeling and simulation in general contexts.

Keywords: System, Modeling, EBE, DES, event-based, engineering, MATLAB, Simevent.

1. INTRODUCTION

Modeling is commonly used for understanding, engineering, and operations of systems [1]. Simulation tools such as DEVSJAVA [2], SimEvent [3], and Ptolemy [4] can be used for modeling complex discrete systems. Each provides its own unique approach for model specification and simulation visualization. A key capability for simulation studies is automated design of experiments such that models can be easily chosen and their simulated dynamics monitored at run-time [5]. Owing to the modular and hierarchical modeling blocks, as well as the simulation-based analysis capability in MATLAB Simevent, the discrete event system specification (DEVS) formalism and its variations have been used in many application of engineering (such as hardware design, hardware/software codesign, communications systems, manufacturing systems) and science (such as biology, and sociology). In this work, Simevents DES presented, modelling and simulation are discussed in general. A Simevent reengineered data center communication network is modeled using DEVS in MATLAB Simevent and we discussed process model realization for the key modules that drives the system. Performance analysis of the system QoS parameters was shown as an accurate scheme of validating the model and finally we present our conclusions.

1.1 Motivation (Simevents DES)

SimEvents is a discrete event simulation tool developed by MathWorks [3]. It adds a library of graphical building blocks for modeling queuing systems to the Simulink environment. It also adds an event-based simulation engine to the time-based simulation engine in Simulink [6]. This work seeks to develop a DES process model which shows the internal behavior of processes in system model using Simevents. Some common uses of Simevents as outline in [6] includes:

i. Process/Logistics simulation

SimEvents is used to model process flows and logistics in order to understand resource availability, inventory management techniques, and the effects of arbitrary events on a mission critical network design. System capacity planning and production planning can simulated while ascertain its optimal performance. A SimEvents model can be

*Corresponding Author: Udeze Chidiebele .C, Electronics Development Institute (FMST-NASENI) Awka Nigeria.

used to estimate through-life support (TLS) requirements such as the size of buffer switches and provisioning of services. A SimEvents model can also predict characteristics such as availability, serviceability and end-to-end latencies based on a specific maintenance schedule [7].

ii. Modeling Performance of a System Architecture

SimEvents is used to build a discrete-event model of the architecture of an electronic system. This architecture model simulates the transactions between different components such as processors, memory, or a communication bus. Simulation of this model provides plots of performance characteristics such as throughput, latency, packet loss and utilization. Systems engineers can use this model to understand the constraints on various shared resources in the electronic system [8]. <http://en.wikipedia.org/wiki/SimEvents>

1. Related Works

In the work carried out in [1], simulation model and complexity evaluation in SimEvents when compared with DEVS-Suite and Ptolemy II simulators shows that Simevents have a more user friendly interface than others and also add to visual complexity metrics in graphical plots. Visual complexity metric is defined as the numbers of components that are displayed for a given model. Again, monitoring inputs and output components in DES models are much easier with Simevents than other tools. Figure 1 shows a typical assembly plant process model in DES presented in[9].

Mathworks [3] presents SimEvents as an extension to MATLAB Simulink with tools for discrete-event simulation of the transactions between components in a system architecture. The author in [12] discussed modeling and simulations from a tutorial perspective, while the work in [13] presented a simevent based transport system model (rail yard) that allows a simulation movement of three trains as discrete events. It explained that discrete-event simulation contains discrete items that are called entities. Each entity can carry data known as attributes. Every passing of entity through block is understood as event that represents immediate discrete incident. This incident changes state of variables, outputs or occurrence of next events.

Also, the architecture of SimEvent and its functionality was presented in [10], while Figure 2 highlights the main functional components of the overall architecture. As a DES simulation engine, SimEvents is driven by an Event Calendar where all future events to occur are listed in ascending order of their scheduled time. SimEvents always processes the first event in this list and updates the DES state accordingly. When such an event takes place, the Cooperative Event Driver is responsible for translating it into a Simulink signal which the Data Exchange module passes on to Simulink so that it may trigger a time driven process or update various model parameters. Conversely, as a time-driven process evolves under the control of Simulink, it may generate events in the form of level-crossing points (from above, from below, or either) that the Data Exchange module appropriately translates so they may be processed by SimEvents blocks. The most challenging aspect of coordinating time-driven and event-driven dynamics is that of proper timing. In the architecture of Figure 2, the system clock is maintained by Simulink and the Cooperative Event Driver is responsible for ensuring consistency between Simulink blocks and SimEvents blocks which interact with the Event Calendar. Essentially, when a pure DES is simulated, the only interaction between SimEvents and Simulink is a simple link to the system clock through the Cooperative Event Driver which ensures that the sample times applied are consistent with times in the Event Calendar. However, the work in [13] forms the basis for our model.

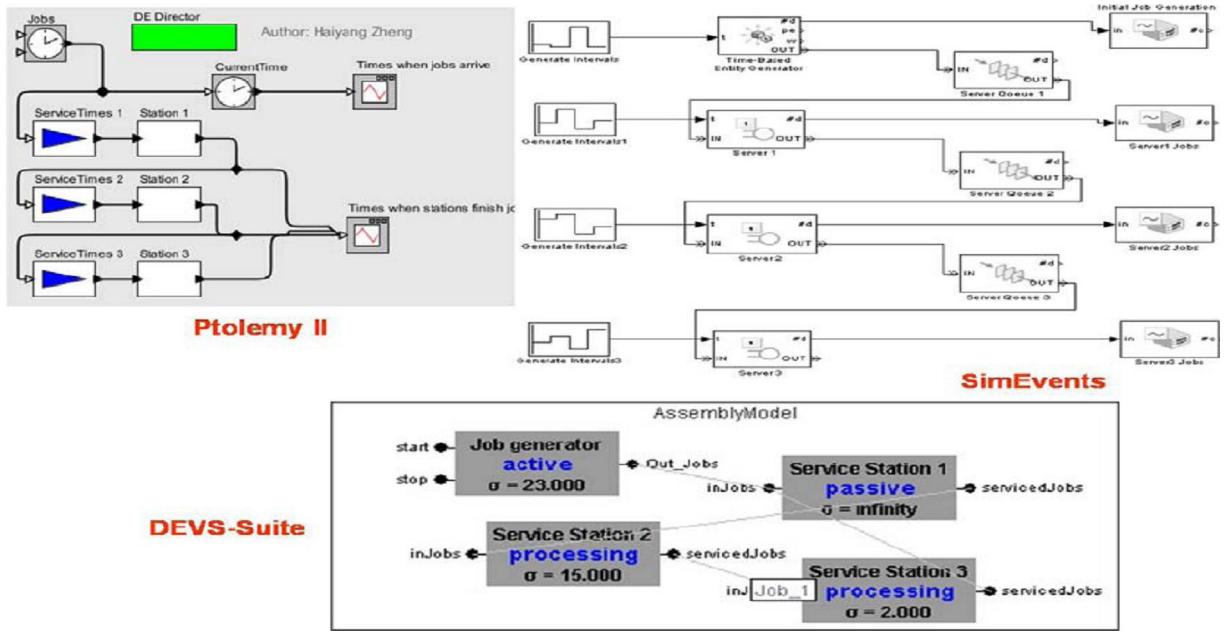


Figure 1: Process Models of an Assembly Line System in DEVS-Suite, Ptolemy II, and SimEvents Simulators [9]

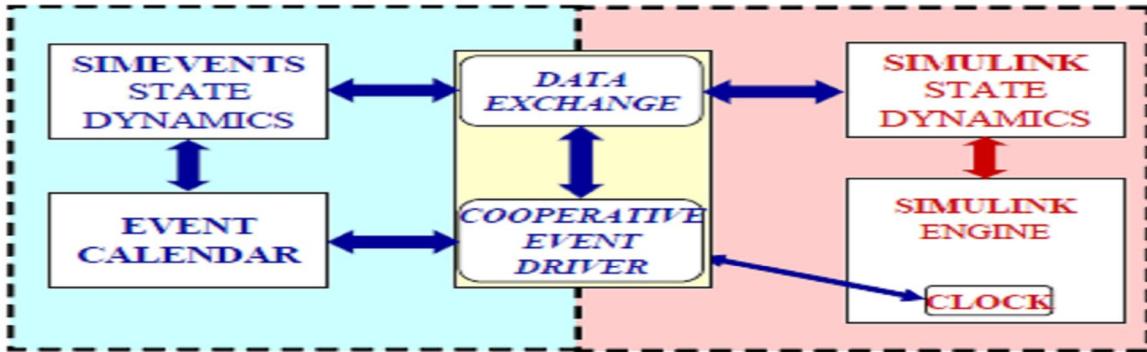


Figure 2: SimEvents and Simulink collaborative functionality [10]

1.1 Modelling and Simulation

Modeling is defined as the process of producing a model and argues that a model is a representation of the construction and working of some system of interest [12]. A model is similar to but simpler than the system it represents. The major objective of a model is to enable an expert predict the effect of changes to the system. In this case, the model should be a close approximation to the real system and incorporate most of its salient features as well as not being unnecessarily complex such that it is impossible to understand and experiment with it. A good model is a meticulous tradeoff between objectivity and simplicity. In all cases, this paper recommends an iterative approach in modeling complex event based systems so as to monitor system performance effectively. Model validity techniques such as simulation allows for comparison between input conditions and model output states with respect to the system output.

Generally, the performance characteristics of a model intended for a simulation study can be developed with the MATLAB simevent software. A model workflow is presented in figure xx. Mathematical model classifications usually includes deterministic models (input and output variables are fixed values) or stochastic models (at least one of the input or output variables is probabilistic); static models(time is not taken into account) or dynamic models (time-varying interactions among variables are taken into account).

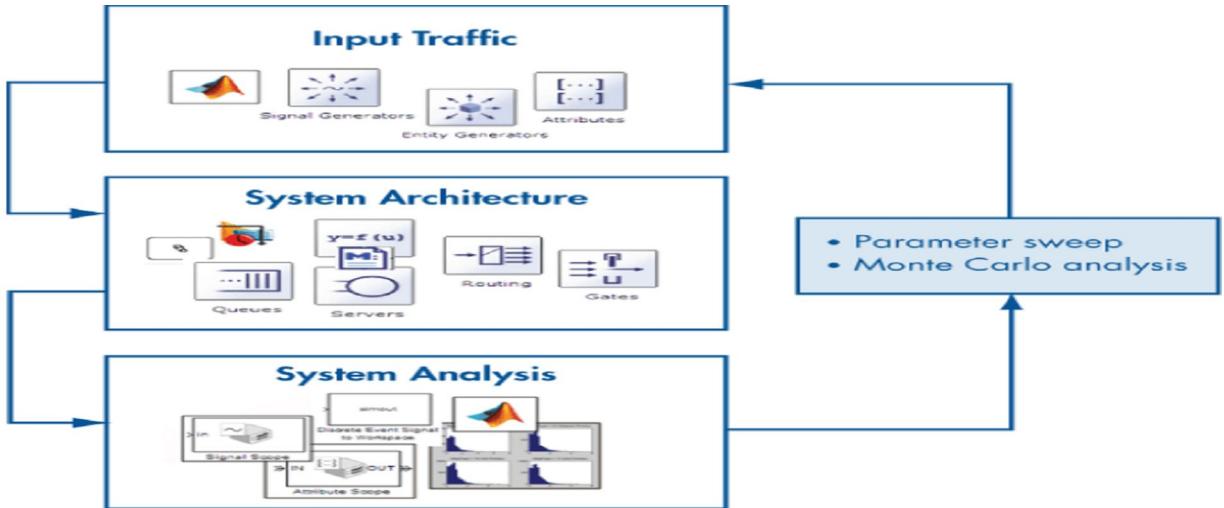


Figure 3: A Workflow Framework in SimEvents to visualize the performance characteristics Of a system. [10]

Typically, simulation models are stochastic and dynamic [12]. This prompts the question: What is Simulation? A simulation of a system model is literally the envisioned operation of a model of such a system. In a technical sense, simulation is a technique used to evaluate the performance of a system (existing or proposed) under different configurations of interest over long periods of real time. Simulation is used before an existing system is altered or a new system built, to reduce the chances of failure to meet specifications, to eliminate unforeseen bottlenecks, to prevent under or over-utilization of resources, and to optimize system performance [12].

Basically, the model can be reconfigured and experimented with, but usually, this is impossible, too expensive or impractical to do in the system it represents owing to obvious factors.

Besides, if the operation of the model can be clearly studied, hence, the properties concerning the behavior of the actual system or its subsystem can be inferred. Consider these case questions:

- i. What is the best design for a new telecommunications network?
- ii. What are the associated resource requirements?
- iii. How will a telecommunication network perform when the traffic load increases by 50%?
- iv. How will a new routing algorithm affect its performance?
- v. Which network protocol optimizes network performance? What will be the impact of a link failure?

This work presents discrete event simulation (DES) as a flexible approach to modeling event based models, a case for a reengineered DCN model. We argue that (DES) will clearly address the aforementioned questions. The central assumption here is that the system changes instantaneously in response to certain discrete events.

1.2 Framework For Simulation Model

In developing a simulation model of any kind, figure 4 becomes very handy to consider. The steps involved in developing a simulation model, designing a simulation experiment, and performing simulation analysis are:

- Step 1. Problem Identification.
- Step 2. Formulation the problem in Clear Terms.
- Step 3. Collection and processing of a real system data.
- Step 4. Formulation and development of a Conceptual model.
- Step 5 Validation of the Model.
- Step 6. Model Documentation
- Step 7. Selection of Appropriate experimental design.
- Step 8. Establishment of experimental conditions for runs.
- Step 9. Execution of simulation runs.
- Step 10. Interpretation and presentation of results.
- Step 11. Recommendation of further course of action

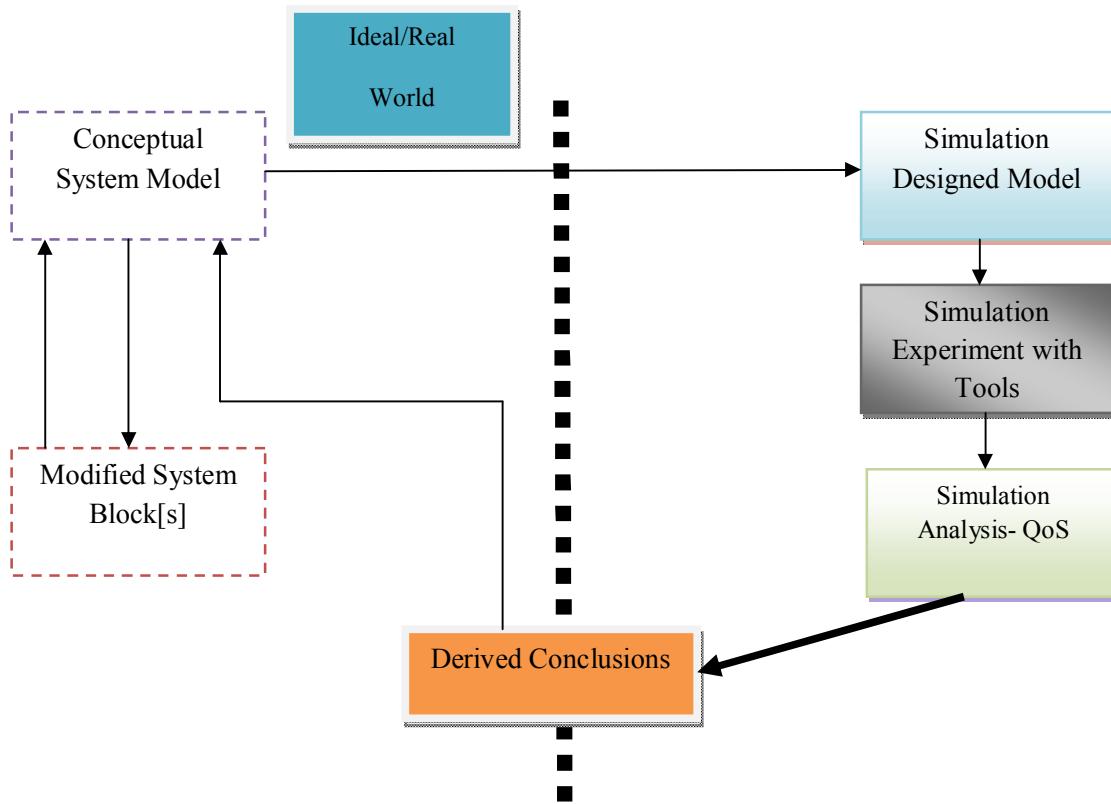


Figure 4: A Generic Simulation Framework

In Simulink, communication across blocks is based on signals only whereas in SimEvents, it is based on both signals and entities. Essentially, SimEvents consists of a number of libraries containing blocks with different system functionalities. The main libraries are the following:

1. Generators: Blocks which generate entities, or function calls (i.e., events that call Simulink blocks), or random varieties.
 2. Queues: Blocks where entities can be temporarily stored while waiting to access a resource.
 3. Servers: Blocks that model various types of resources.
 4. Routing: Blocks that control the movement of entities as they access queues and servers.
 5. Gates: Blocks that control the flow of entities by enabling/disabling access of entities to certain blocks.
 6. Event Translation: Blocks that enable communication between SimEvents and Simulink by translating events into function calls.
 7. Attributes: Blocks that assign and modify data to entities. Various control actions are then made based on the values of these data, allowing blocks to differentiate between entities they process.
 8. Subsystems: These allow a combination of blocks to be executed upon occurrence of specific events (not upon Simulink sample times).
 9. Timers and Counters: Blocks that measure event occurrence times or time elapsing between events, and blocks that count occurrences of particular event types. These data are supplied to standard display or scope blocks in Simulink or specialized scopes designed specifically for SimEvents. Figure 5 shows a generic DES Simevent library. In this work, SimEvents has been successfully used to model two aspects of distributed system for our reengineered DCN:
- (i) Delays in message delivery due to loading of busses and networks.
 - (ii) Delays in task execution due to processor loading.

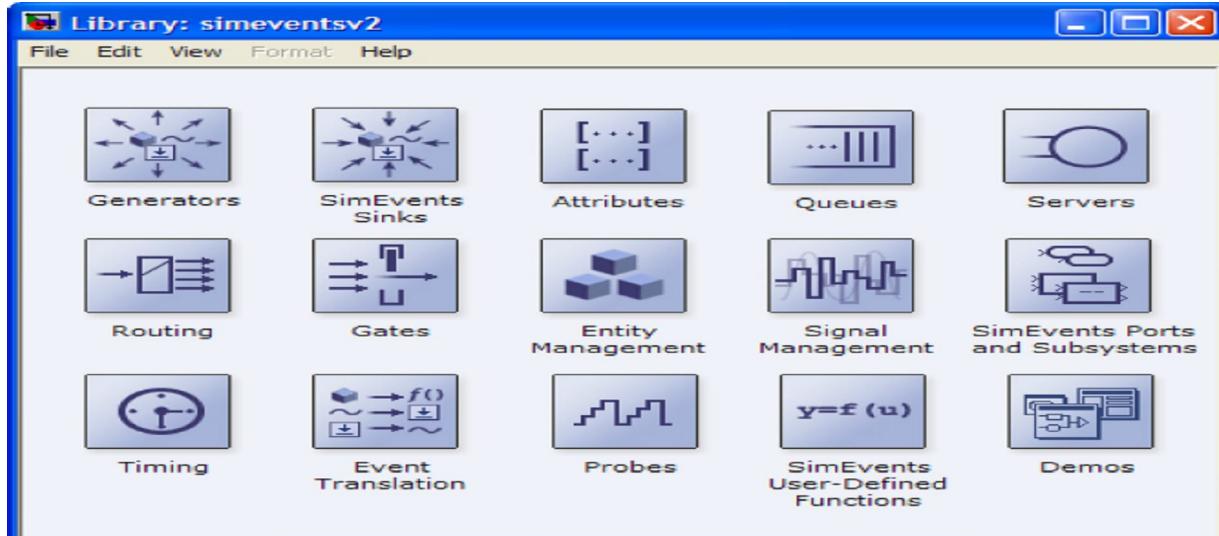


Figure 5: SimEvents block library for building discrete-event simulation models [10].

2. MATLAB Simevent Implementation in Reengineered DCN

By leveraging on the capabilities offered by MATLAB SimEvents such as its graphical drag-and-drop interface for building a discrete-event model, in our implementation approach, the key features explored includes:

- Entity-based modeling for representing packets, tasks, and parts
- Attributes for attaching scalar, matrix, and complex data to packets, tasks, and parts
- Libraries of predefined blocks, such as queues, servers, routing, and generators, for modeling system architecture and process flow diagrams
- Built-in statistics such as delay, throughput, and average queue length
- Library block for writing algorithms to customize operations such as routing, processing delays, and prioritization
- Entity and attribute aggregation blocks for modeling data hierarchy
- Hybrid simulation capabilities for models that contain both event-based and time-based components

Simevents integration with MATLAB allows customization of the process flow in our implementation model. In this context, a MATLAB function can be developed to represent a task-scheduling sequence, switching and routing of traffic in a process flow model. With SimEvents and MATLAB Integration within the same tools environment, it allows for a flexible generation of custom random distributions of input tasks as well as custom data statistics from a SimEvents model using a MATLAB command line interface.

2.1 Process Model for Reengineered DCN

This paper develops a discrete event based process model and simulates a reengineered data center communication network comprising of four data center LAN nodes with an inclusion of a virtualization server for creating virtual instances of resources and applications for DCN nodes. A T-junction link modeled for 10GPs bandwidth was used. The MAC Controller block implements the CSMA/CD +TAMP (Carrier Sense Multiple Access with Collision Detection and Traffic Arbitration Message Protocol) Mechanism. This access technique of MAC bus system makes for an efficient and flexible data throughput while optimizing bandwidth in the reengineered DCN model. MATLAB Simevent was employed in the model. Also, the CSMA/CD+ TAMP was modeled into the DCN MAC controllers for full duplex traffic flow while processing queues (frames) in such a manner as to avoid packet losses and congestion in the DCN when scalability consideration arises. Figure 6 shows the process model architecture developed in this paper.

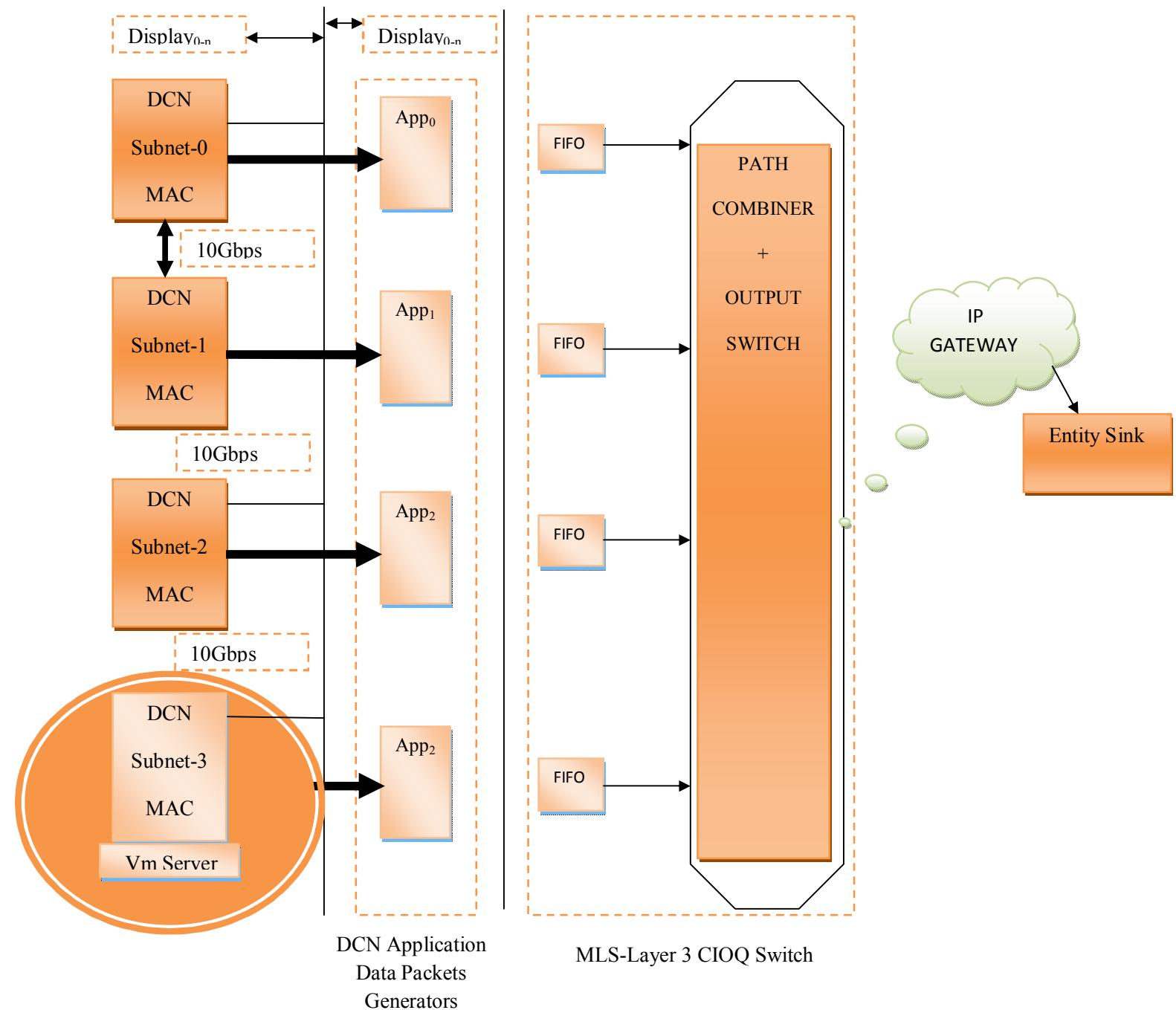


Figure 6: The Process Model Architecture

3.2 Model Architecture

The process model architecture shown figure 6 was implemented with MATLAB Simevent as shown in figure 7. The top level model which includes four subnets DCN LAN nodes, each consisting of the following (as shown in figure 6):

- An Application data packet block that models the DCN source data with its interface to a switch and router gateway.
- A MAC (medium access control) controller that governs the DCN's use of the shared channel and link resources.

- A T-junction that connects a LAN node to other DCN subnets networks.
- Switching Path Combiners, Gateway IP Clouds and Sinks

The packet generation rate and packet size range are specified at the Application data packet blocks, the transmission buffer size specified at the MAC Controller blocks, and the length of the cable specified at the Cable blocks. The terminator, T-junction, and cable blocks at the bottom of the model represent physical components of the network. Figure 7 shows the simevent model.

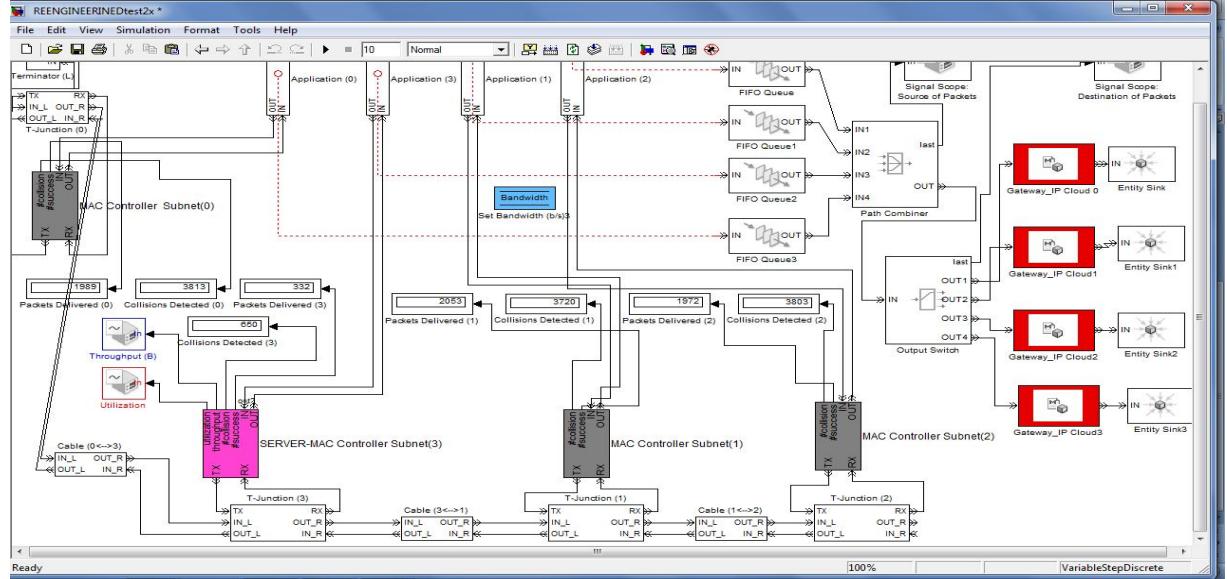
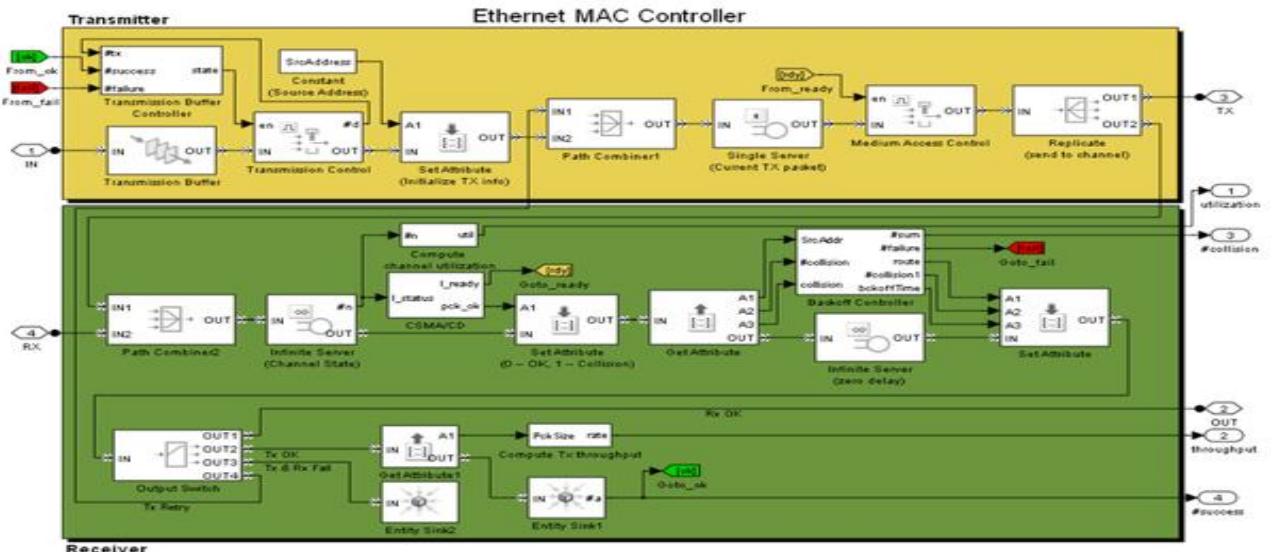


Figure 7: Reengineered DCN Process Model with Simevent

Using MATLAB Simevent, we developed figure 6 as standard DCN communication model that uses a CSMA/CD+TAMP protocol to manage use of the shared channel for high speed traffic computation. Each DCN MAC controller node physically monitors the traffic on the channel and initiates a transmission only when the controller detects no other traffic on the channel and issues TAMP to other DCN nodes. With this access technique, channel contention is less likely to occur as transmissions from multiple DCN nodes are in excess full duplex mode. Packets collision with other packets can make a fixed number of additional attempts after its short waiting period elapses, using the binary exponential backoff algorithm. In this model, the CSMA/CD+ TAMP protocol is implemented in the MAC Controller subsystems shown in figure 8.



3. Performance Evaluations

This model includes various visualization that capture the performance of the DCN Ethernet bus system:

- Display blocks that show the number of packets that each DCN node observes as successful or unsuccessful communications. Propagation delay modeled in the Cable blocks can cause different nodes to observe slightly different results.
- The throughput plot shows the transmission throughput at the DCN server
- The server utilization plot shows the proportion of time that the server is in use allocating resources.

We have evaluated the proposed reengineered DCN process model using MATLAB Simevent Simulator [3]. The work confirms that the packet transmission from sources to destinations have effective throughput, latency, and efficient server utilization. Figure 9 shows our simulation runs for poisson process distribution. Before the simulation runs, link consistency and other block set configurations were verified. The simulation start time was gradually increased from rest (0) to 1000. For each run, the transmission buffer sizes, packet generation rate, packet sizes, FIFO capacity are all tuned. Finally, throughput, latency, utilization, stability and convergence tests were carried out. Now, we are compiling the data values for our QoS parameters and will report our experimental evaluation results in the near future work with the plots for the QoS parameters. Figure 10 and figure 11 are the graphical plots for server MAC controller while Figure 12 depicts the Simevent State flow Chart for Ethernet Controller in this paper.

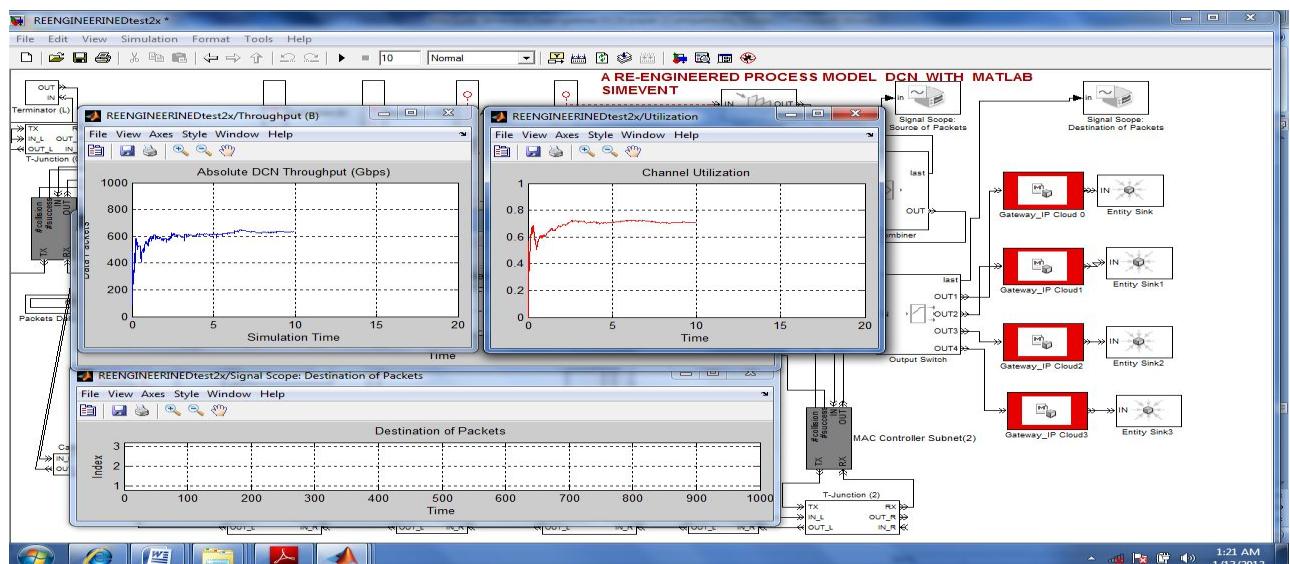


Figure 9: A Simulation run on MATLAB Simevent

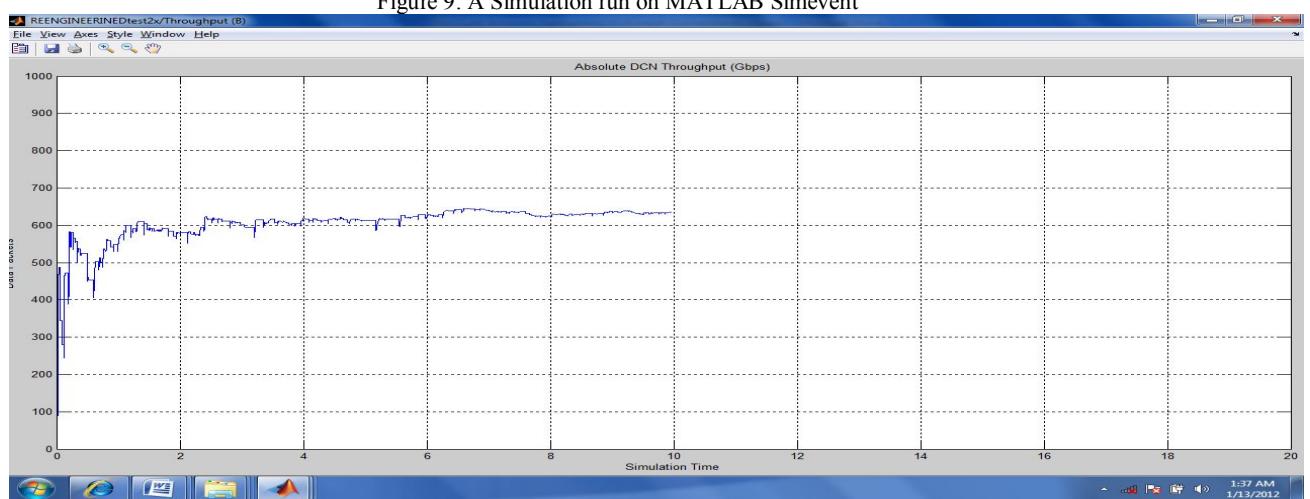


Figure 10: Absolute DCN throughput Plot

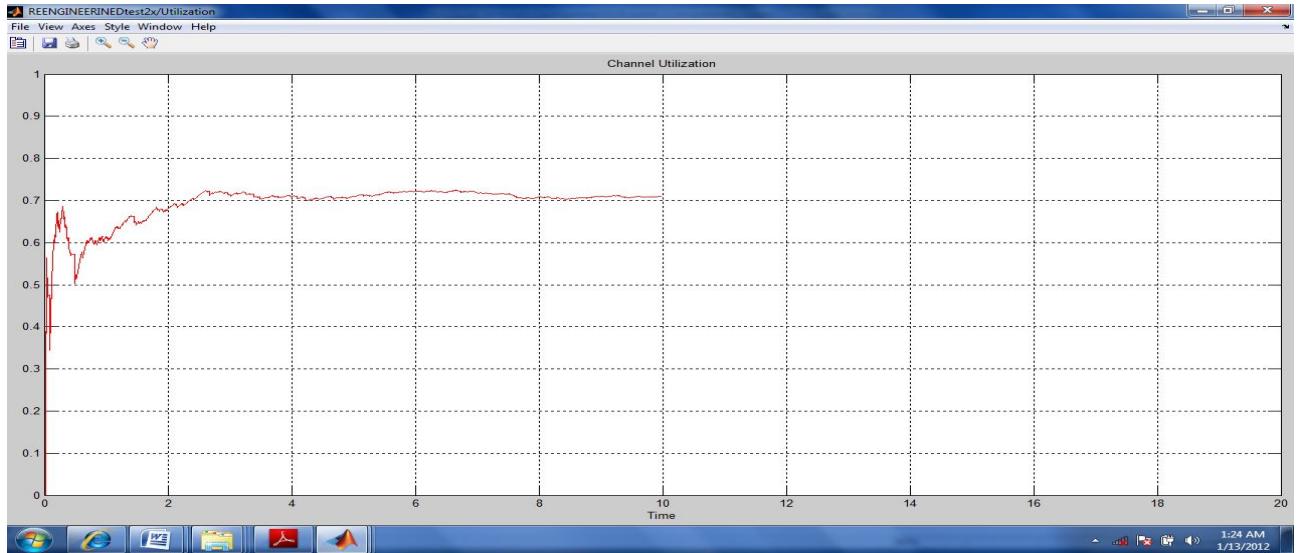


Figure 11: Server Channel Utilization Plot

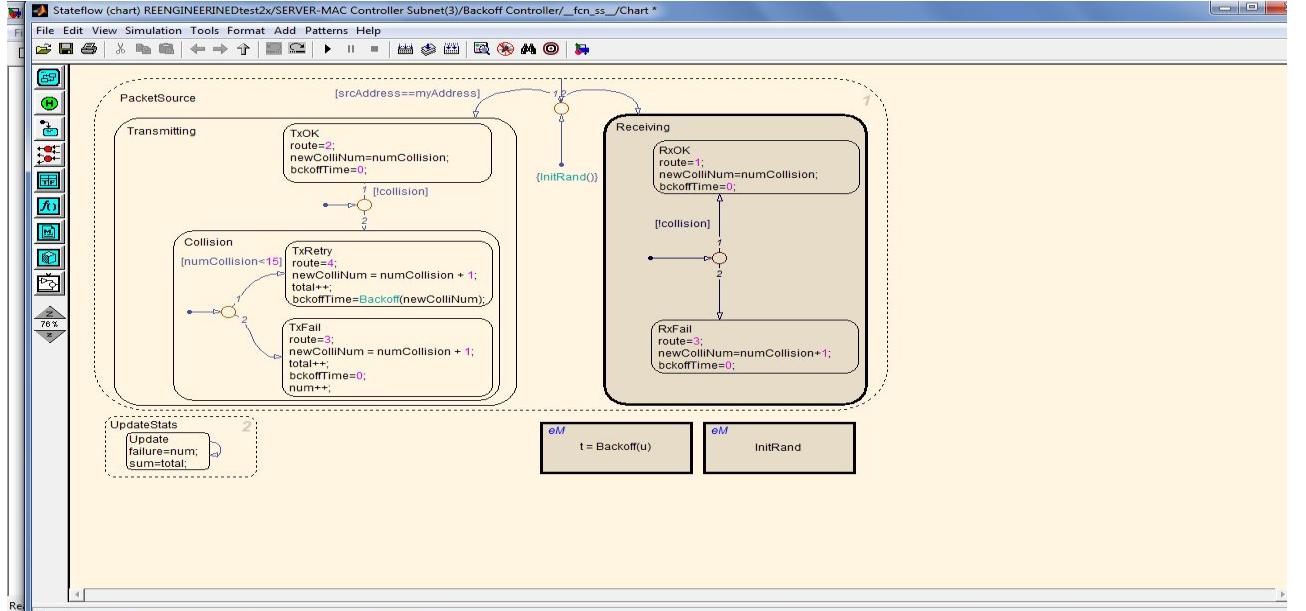


Figure 12: Simevent State flow Chart for Ethernet Controller

4. Conclusion

In this paper, we have presented MATLAB Simevent approach for developing the process model of a reengineered DCN that scales well in today's high speed DCN computing. The developed models provides end to end reliable packet delivery in a short time with CSMA/CD+TAMP protocol that regulates losses and enhances throughput in the DCN. A discussion on modeling and simulation is also presented in this work. The simulation plots for absolute throughput and server utilization shows that load intensities and buffer sizes of the DCN MAC controllers needs to be adjusted to higher values for optimal performance. Fast convergence, fairness, efficient throughput, and utilization are QoS of interest in this work. Though the simulation presented here is just for one case, it's obvious that by carefully selecting the DCN block parameters, Ethernet packet generators, switch FIFO capacities and iteratively running the simulations, a more stable system will be generated. Future work will present modifications and complete data plots for the selected QoS parameters while justifying the system stability.

REFERENCES

1. Sungung Kim, Hessam S. Sarjoughian, Vignesh Elamvazhuthi, "Devs-suite: A Simulator Supporting Visual Experimentation Design And Behavior Monitoring", (Unpublished).
2. Arizona Center for Integrative Modeling and Simulation 2007. <http://www.acims.arizona.edu/> SOFTWARE.
3. Mathworks. 2007. <http://www.mathworks.com>
4. Lee, E. A. 2003. Overview of the Ptolemy Project. University of California, Berkeley, CA, USA.
5. Dalle, O. 2007. An Instrumentation Framework for Component-Based Simulations based on the Separation of Concerns Paradigm. 6th EUROSIM Congress, Ljubljana, Slovenia.
6. Cassandras, Christos (2007-11-27). "Introduction to Discrete-Event Simulation". Springer US. <http://www.springerlink.com/content/g82w56/?v=editorial>. Retrieved 2009-11-03.
7. Bender, Alex (2009-13-07). "Effects of decay uncertainty in the prediction of life-cycle costing for large scale military capability projects". 18th World IMACS / MODSIM Congress, Cairns, Australia. <http://www.mssanz.org.au/modsim09/E1/bender.pdf>. Retrieved 2009-11-04.
8. Apte, Anuja (2008-03-01). "Modeling System Architecture and Resource Constraints Using Discrete-Event Simulation". MATLAB Digest - March 2008. http://www.mathworks.com/company/newsletters/digest/2008/mar/discrete_event.html. Retrieved 2009-11-04.
9. Elamvazhuthi, V. 2008. Visual Component-Based System Modeling with Automated Simulation Data Collection and Observation, MS Thesis, Department of Computer Science and Engineering, Arizona State University, Tempe, AZ, USA
10. Michael I. Clune and Pieter J. Mosterman, Christos G. Cassandras, "Discrete Event and Hybrid System Simulation with SimEvents"
11. Anu Maria, " Introduction To Modeling And Simulation" Proceedings of the 1997 Winter Simulation Conference ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson.
12. *K. Valigura, M. Foltin, M. Blaho*, "Transport System Realization in Simevents Tool"(Unpublished)