

## PRODUCTION SYSTEM MODELING AND SIMULATION USING DEVS FORMALISM

DARÍO AMAYA HURTADO<sup>1</sup>

RICARDO A. CASTILLO<sup>2</sup>

✉ ÓSCAR F. AVILÉS<sup>3</sup>

OLGA LUCÍA RAMOS<sup>4</sup>

### ABSTRACT

This article presents the Discrete Event System Specification (DEVS) formalism, in their atomic and coupled configurations; it is used for discrete event systems modeling and simulation. Initially this work describes the analysis of discrete event systems concepts and its applicability. Then a comprehensive description of the DEVS formalism structure is presented, in order to model and simulate an industrial process, taking into account changes in parameters such as process service time, each station storage systems structure and process tasks coupling. For the MatLab® simulation, the Simevents Toolbox was used for theoretical developments validation.

**KEYWORDS:** Discrete event systems; DEVS; SimEvents®.

## MODELADO Y SIMULACIÓN DE UN SISTEMA DE PRODUCCIÓN USANDO EL FORMALISMO DEVS

### RESUMEN

En este artículo se presenta el formalismo de especificación de sistemas a eventos discretos (DEVS), en sus configuraciones atómica y acoplada. Este formalismo es utilizado en el modelamiento y simulación de sistemas que responden a eventos discretos. Inicialmente, en este trabajo se realiza una presentación de los sistemas de eventos discretos y sus aplicaciones. Luego de esto, se realiza una descripción del formalismo, en función del modelo y la simulación de procesos industriales, teniendo en cuenta el cambio en los tiempos de servicio de estaciones que trabajan de manera acoplada. Finalmente se presenta un caso de estudio, el cual es modelado y simulado utilizando la herramienta Simevent de MATLAB®.

**PALABRAS CLAVES:** Sistema de eventos discretos, DEVS, SimEvents®.

---

<sup>1</sup> Electronic Engineer. Industrial Automation Specialist. Master in Teleinformatics. PhD in Mechanical Engineering. Researcher Professor, Universidad Militar Nueva Granada .

<sup>2</sup> Mechatronics Engineer. Master in Mechanical Engineering. PhD Candidate in Mechanical Engineering. Researcher Professor, Universidad Militar Nueva Granada.

<sup>3</sup> Electronic Engineer. Electronic Instrumentation Specialist. Master in Automated Production Systems. PhD in Mechanical Engineering. Postdoctor in Mechanical Engineering. Researcher Professor, Universidad Militar Nueva Granada.

<sup>4</sup> Electronic Engineer. Electronic Instrumentation Specialist. Master in Teleinformatics. Doctoral Candidate of Engineering. Researcher Professor, Universidad Militar Nueva Granada.

✉ *Correspondence author: Avilés, O.F. (Óscar Fernando). Universidad Militar Nueva Granada, Cr. 11 No 101-80. Dpto. Ing. Mecatrónica, Bogotá (Colombia). Tel.: (571) 6500000 Ext. 1297.  
Email: oscar.aviles@unimilitar.edu.co*

*Paper history:*  
Paper received on: 27-II-2013 / Approved: 1-VIII-2014  
Available online: December 30 de 2014  
Open discussion until December 2015

# MODELAGEM E SIMULAÇÃO DE UM SISTEMA DE PRODUÇÃO USANDO O FORMALISMO DEVS

## RESUMO

Em este artigo, é apresentado o formalismo de especificação de sistemas a eventos discretos (DEVS) em suas configurações atômicas e acoplados. Este formalismo é usado em modelagem e simulação de sistemas que respondem a eventos discretos. Inicialmente, em este trabalho, é realizada uma apresentação dos sistemas de eventos discretos e suas aplicações. Depois disso, é feita uma descrição do formalismo, em função do modelo e da simulação e de processos industriais, tendo em conta a variação nos tempos de serviço de estações que trabalham de um modo acoplado. Finalmente, se apresenta um caso de estudo, o qual é modelado e simulado utilizando a ferramenta Simevent de MATLAB®.

**PALAVRAS-CHAVE:** Sistema de eventos discretos; DEVS; SimEvents®.

## 1. INTRODUCTION

Developments in technology, reflected mainly in computer systems, communication systems and manufacturing systems, have resulted in new methodologies for modeling and analysis related to systems that respond to discrete events.

Such systems are characterized by performing tasks in real time, high efficiency, and accuracy. Thus, requiring systematic methods associated with computational tools to support the analysis and simulation of these systems (Capocchi, L., *et al.*, 2006).

In this case, the simulation is essential, to understanding the behavior and operation of the system. Several formalisms were developed for the analysis and simulation of discrete events, the DEVS formalism being one of the most popular among them (Filippi, J.-B., and Bisgambiglia, P., 2004; Hong, K. and Kim, T., 2006).

One example is its application in industrial manufacturing systems, which are defined by Palaniappan, S., *et al.* (2006) and Nikolaidou, M., *et al.* (2006) as an activity in charge from the processing of raw materials into finished products ending with the distribution. This transformation includes product design, materials selection, process planning, purchasing, production control, product support, marketing, sales, product distribution, and customer service, among others. This type of systems consists of cells controlled by discrete events.

An event may correspond, for example, with a particular task completion in a production system, the arrival and departures of customers, a package transmission through a communications system, requests arrival and its fulfillment, machine failure, the arrival and departure of parts, signals connection and disconnection by the process operator, among others.

A discrete event system (DES) is characterized by having a dynamic evolving accordingly to the abrupt physical events occurrence. These events usually happen asynchronously in time. The events stimulate the dynamics causing a state transition system (Bergero, F., 2000). Today with the changing market demands and economic globalization, there are expectations for highly flexible production systems, able to take advantage of new information technologies and changes in demand and event-based modeling offers the possibility of representing the dynamic behavior of processes.

The acquisition of the process model is an important element in the control systems realignment. For this purpose, several formalisms were developed that allow the modeling of discrete events variables. Among the formalisms commonly used in modeling, analysis and control of discrete event systems (Van der Schaft A. and Schumacher H., 2000) are the Max-min-plus-scaling (MMPS) systems, the Automata, Petri Nets, Discrete Event System Specification (DEVS), and Grafset

This article focuses in the DEVS formalism used for the discrete systems modeling and simulation applied to industrial processes using the Matlab® computational tool as simulation support. This paper is structured as follows: the first section presents an introduction to the Discrete Event Systems; the next section presents both the DEVS atomic formalism and the coupled DEVS structural descriptions. The third section describes an industrial process modeling containing discrete events variables, with the three workstations attached. The fourth section presents the manufacturing process simulation illustrated in the third section, for the event the Matlab® Simulink®, Simevents® tool was used, taking place in the fifth section are the conclusions and lastly the references and bibliographic sources.

## 2. DESCRIPTION OF THE DEVS ATOMIC FORMALISM

For a long time, mathematical tools have been used to represent the behavior of systems governed by time or physical phenomena as differential equations. Currently, and taking into account the technological developments made by man, such as computers, transportation systems, manufacturing, communications and others. Moreover, considering that their behavior is governed by events occurring asynchronously in time (Van der Schaft A. and Schumacher H., 2000).

This system is typically called discrete event (DES), and its analysis is very complex, due to a series of requirements such as, copyrights issues, productivity requirements, limitations in response time and so on Hong, K. and Kim, T. (2006). In 1972, the mathematician Bernard Zeigler proposed a formalism to represent the DES (Villani E., et al., 2007), which was called DEVS.

Branicky M. (1995) describes the DEVS formalism as a conceptual environment that specifies the model, and the simulation of discrete event systems. Modeling can be done through the system decomposition into smaller models, specifying the coupling between them.

The smaller models are defined as the atomic model, the fundamental element, representing the processing "molecular" unity, and the second is the coupled model (Alur, R., et al., 2000; Kain, S., et al., 2008).

The atomic model can be defined as (1)

$$H = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, t_a) \quad (1)$$

Where:

$X$  : Is the input events set of values.

$Y$  : Is the output events set of values.

$S$  : Is the set of state values

$\delta_{int}, t_a$  : It is the internal transition and the time progression functions, respectively

$\delta_{ext} : Q \times X \rightarrow S$  : Is an external transition function, where  $Q$  is the set of states

$$M = \{(s, e) | s \in S \wedge 0 \leq e \leq t_a(s)\} \quad (2)$$

$\lambda : S \rightarrow y$  : Is the output function

$t_a : S \rightarrow \mathbb{R}_{0, \infty}^+$  It is the time progression function, which determines the maximum duration that the system can stay idle. After a time  $t_a(s)$  without the occurrence of external events, the system internally transitions to a new state  $S_2$ , calculated by  $S_2 = \delta_{int}(s)$

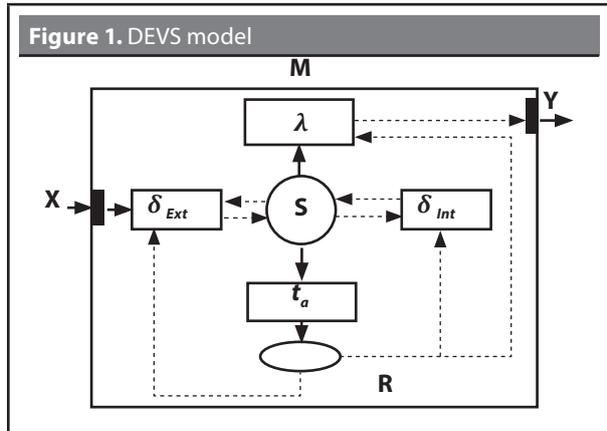
The transition from state  $S_1$  to  $S_2$ , also produces an output event, defined by  $y = \lambda(S_1)$ .

The following equation statement (3) represents the atomic model state equation.

$$q' = \delta_{int}(q) \oplus \delta_{ext}(q, X) \quad (3)$$

Meaning, that system's new state depends on one of the two transition functions, but not on both simultaneously.

**Figure 1** shows the DEVS atomic model semantics (Sikun, L., et al., 2005). The model responds to external input events, according to its external transition function, in the absence of events in a specific time set by the time advance function, the model changes state according to the internal transition function and generates an output event for the other models.

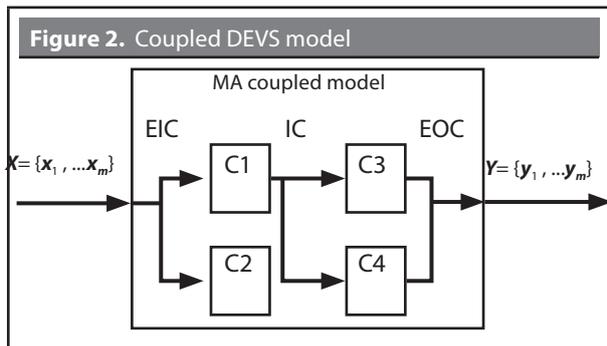


### 3. DEVS FORMALISM FOR COUPLED SYSTEMS

The atomic models can interact through a coupled model. This allows for to the division of a modeling issue from a complex system into small units that can be coupled and make their representation easier (Chen, C., and Yücesan, E., 2001; Nikolaidou, M., et al., 2006). **Figure 3** shows a coupled DEVS diagram.

With its several closed loops, the coupled DEVS modules can, not only be connected in cascading form, but also allows feedback configurations between the atomic models, thus permitting an industrial production system representation.

The coupled DEVS is formed of several atomic models (D), connected through internal link and is defined by the tuple of **Equation 4** and present in the **Figure 2**:



$$CM = \langle X, Y, D, \{M_d \mid d \in D\}, EIC, EOC, IC, select \rangle \quad (4)$$

$X$ : Is the input set, for the reception of incoming evVents, where:

$$p \in Inport, v \in X_p$$

$Y$ : It is the output set, for the transmission of external events.

$$p \in OutPorts, v \in Y_p$$

$D$ : It is the components package (basic model)

$D$ : It is the DEVS model for each  $d \in D$ . Verifiable according to the connections.  $M_d = \langle X_d, S, Y_d, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$ , is the classic DEVS, with several input/output ports.

$$p \in Inports_d, v \in X_p$$

$$p \in OutPorts_d, v \in Y_p$$

$EIC$ : It is the input links that connects the coupled model entries to one or more of its components input areas.

$$EIC \subseteq \{(N, ip_N), (D, ip_D) \mid ip_N \in Inports, d \in D, ip_D \in InPorts_d\}$$

Where:

$(N, ip_N)$ : Represents the coupled model  $ip_N$  input port.

$(d, ip_d)$ : Represents the component's  $ip_d$  input port.

$((N, ip_N), (d, ip_d))$ : Represents the connection between the two ports.

$EOC$ : The set of output links that connects the outputs from one or more components to the coupled model input port.

$$EOC \subseteq \{(d, [(op)]_d), (N, [(op)]_N) \mid [(op)]_d \in Outports, d \in D, [(op)]_d \in [(OutPorts)]_d\}$$

Where:

$(d, op_d)$ : Represents the component's  $op_d$  output port.

$(N, op_N)$ : Represents the composite model's  $op_N$  output port.

$((d, [(op)]_d), (N, [(op)]_N))$ : Represents the connection between the two ports.

$IC$ : It is the internal links set connecting the components' output ports to the coupled models input ports.

$$IC \subseteq \{(a, op_a), (b, ip_b) \mid a, b \in D \text{ con } a \neq b, op_a \in OutPorts_a, ip_b \in InPorts_b\}$$

Where:

$(a, op_a)$ : represents the coupled model's  $ip_N$  port of entry.

$(b, ip_b)$ : represents the d component's  $ip_d$  input port.

$((a, op_a), (b, ip_b))$ : represents the connection between the two ports.

Select: sets event priorities, even if an internal event is already scheduled for the same time.

#### 4. EXPERIMENT

This section presents a study case to illustrate the DEVS formalism application in an industrial production process. The experiment comprises a production line consisting of three workstations, coupled sequentially. See **Figure 3**.

The production line model was based on each workstation's atomic model and their integration produced one coupled model. The first station is pneumatically operated, charged with supplying the raw material into a production line assembling process. The model shows this station as a black box, responsible for producing a consistent product on a support platform by completing the final product. In this manner, it can be displayed as a system with an input buffer, which serves the products to be processed by a server, which represents the delivery of the raw material to the production line.

The raw material from the previous station arrives at the next station that is in charge of the Lego type product assembling on a support platform.

With the product assembling complete, it passes to a third workstation responsible for the product labeling, similar to the previous workstations, this workstation is connected to a server able to perform the designated activity with a product-receiving buffer.

In the modeled system, each station consists of its input buffer associated with the server  $P_x$ , configured in series to represent the sequential process. The raw material to serve as support for assembling is represented by the elements  $ax$ , with an arrival rate linearly independent. The red circles imply that a server is handling an event (busy) and that a product in the queue is waiting for the service.

The model is defined as follows:

The components' serial connections are accomplished as follows: output of the first process connects to the input of the second process, and lastly the output of the second process connects to the input

of the third process. This represents what is called an internal coupling between components.

The corresponding coupled DEVS model composition is as follows:

$CM = \langle X, Y, D, \{M_d, d \mid d \in D\}, EIC, EOC, IC, select \rangle$

$X_{in} = \mathbb{R}$ : Possible values that can be entered by the port  $A_1 = \{a_1, a_2, \dots\}$ .

$InPorts = \{ "A_1" \}$ : Composite model input port

$X = \{ (a_x, v) \mid v \in \mathbb{R} \}$ : Composite model input port.

$OutPorts = \{ "d_x" \}$ : Composite model output port corresponding to the process 3 output.

$Y_{out} = \mathbb{R}$ : possible values that may hold the composite model  $\{d_1, d_2, \dots\}$ .

$Y = \{ (d, x, v) \mid v \in \mathbb{R} \}$ : Composite model output port, and the probable values remaining from the output of previous process.

$D = \{ p_1, p_2, p_3 \}$ : Set of names of composite set correspondent to the atomic models of each process.

$EIC = \{ (N, p, x), (p, 1, a, x) \}$ : It is the input connection between the composite models in each process.

$EOC = \{ (p_3, out), (N, out) \}$ : Output connection from the last process to the composite model output port.

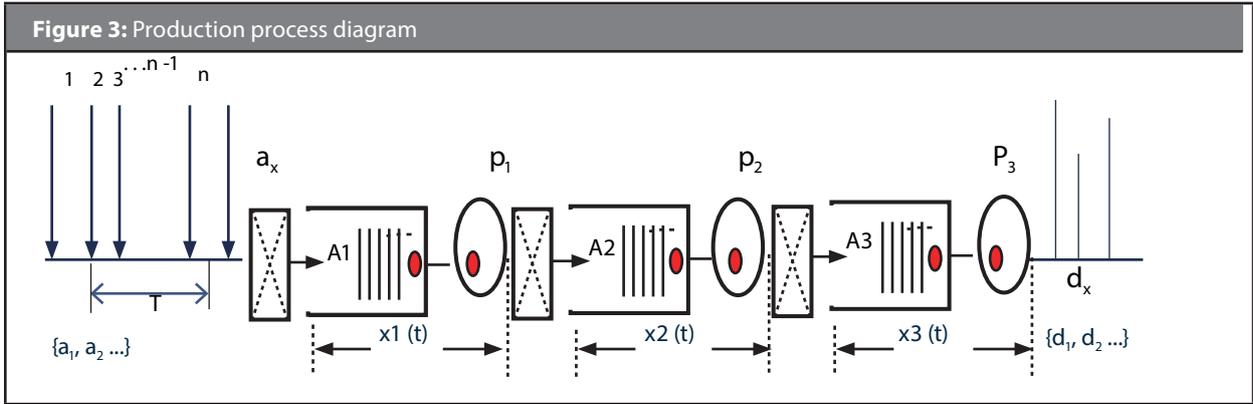
$IC = \{ ((p_3, out), (p_2, a_x)), ((p_0, out), (N, a_x)) \}$

$Select(D')$  = Equivalent to the internal conditions among processes.

Select: equivalent to the faster process selection.

For the model system simulation, the MATLAB® tool and Simevents toolbox were used; these software are geared to the simulation processes involving discrete time variables, i.e., the manufacturing processes can identify bottlenecks in a production line, perform the identification of resources needed for production, execute the analysis of the performance characteristics of the system, response times and production delays, among many other behaviors that can be analyzed.

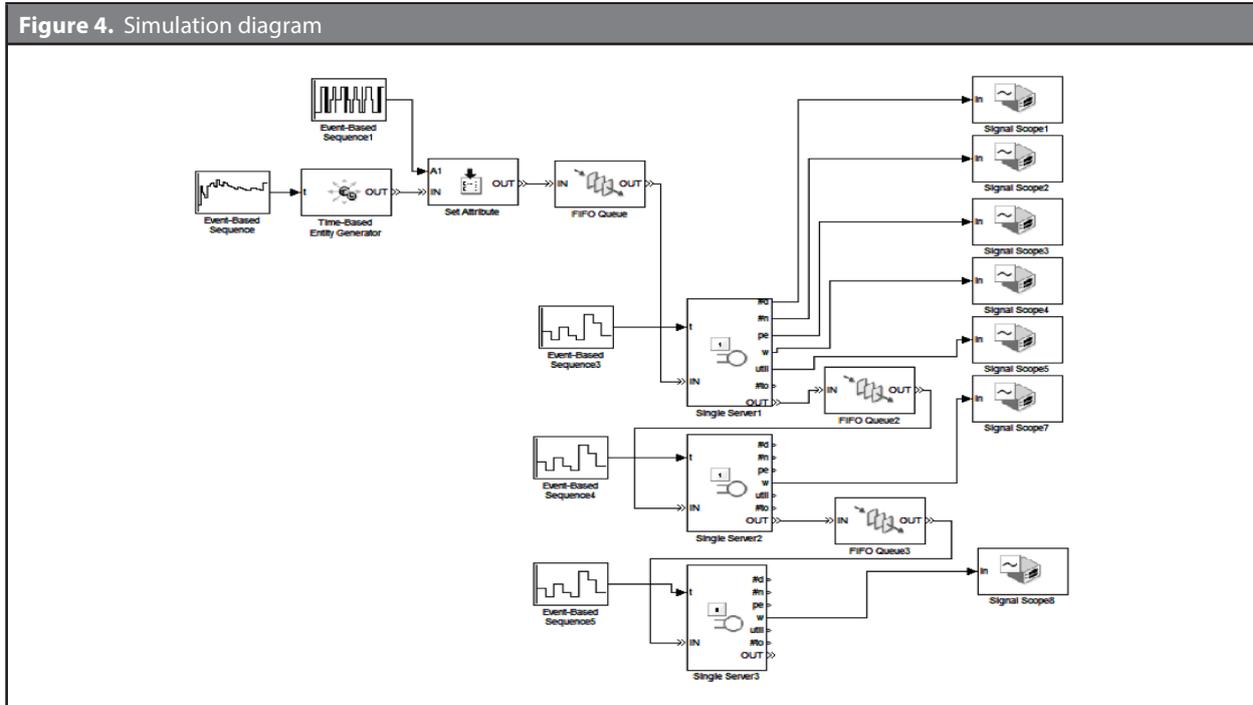
The Simevents®, "Event-Based Sequence", allows the generation of a signal of events based on data drawn by the user. With this block, a date is generated with all the Vs that arise in a server.



The "FIFO Queue" block simultaneously has  $N$  entities and provides an entity through its output port, yet it retains said entity if the output port is blocked, and other entities are still available. When keeping several entities in storage, the block will always deliver the first that was stored, i.e., the first entity to be stored is the first to exit through its output port.

The Single Server block, serves an entity for a period and then delivers an entity to its output port. In case of a blocked output port, the entity will wait in the block until the port become available. In this application, the servers represent each of the workstations in the production line. This block input ports

are represented by IN, which is the entrance way for the entities to be served,  $t$  is the service time defined in seconds, for an entity's new arrival time. The outputs are divided into two blocks; the first is "Entity Output Ports", composed of OUT, which is the exit period after the completed task. P is the port for entities displaced by the arrival of a higher priority entity. TO is port for entities with wait time; #  $d$  is the number of entities already tended by the server; #  $n$  is the number of entities that are currently on the block;  $w$  shows the entities mean waiting time, useful for showing the server usage time; # displays the number of entities that have exhausted their time.



The graph representing the industrial production model dynamics is shown in **Figure 4**. It can be observed how each server represents the dynamics of each of the stations involved in industrial production system, activities performed in each of the three stations make up a production line, each server has a random processing time, which simulates the time it requires the station to carry out their activity. Furthermore, each station has a storage buffer containing the entities prior to being processed.

The simulation's purpose is to perform the analysis of the dynamics relating to the times of use, mean waiting time, number of units serviced at the stations.

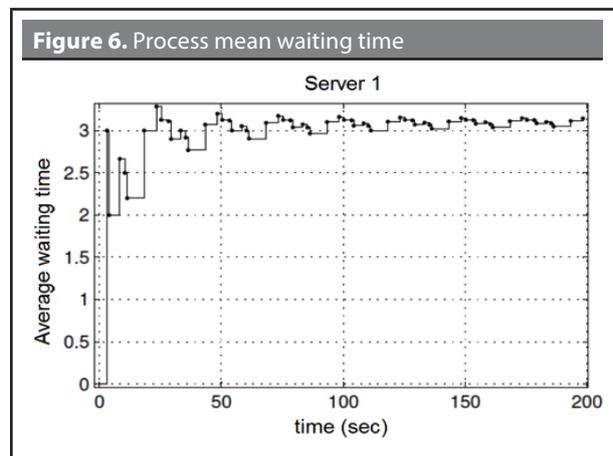
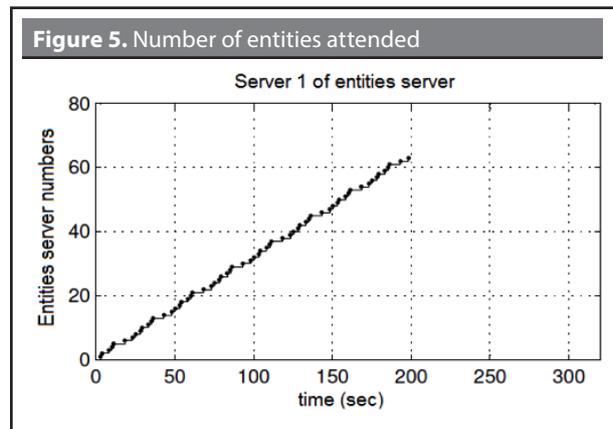
## 5. RESULTS

According to the results on the behavior of each of the workstations, analysis was carried out only for workstation one. This is due to the similarity between results obtained from the other workstations.

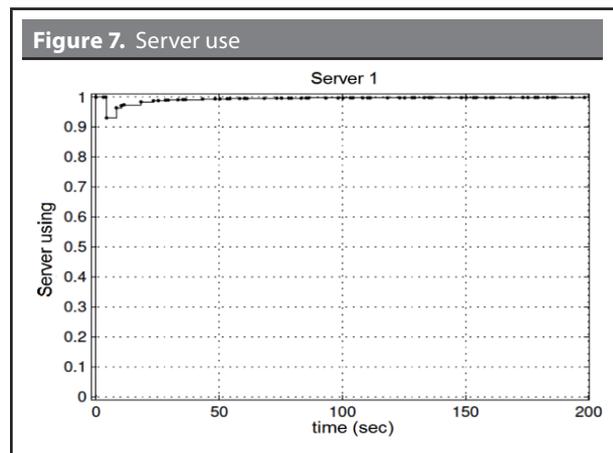
Results were obtained for, mean waiting time, usage time, and number of units serviced at the stations.

In **Figure 5** displays statistics from the server one, representing the workstation with raw material input. The generator delivers the units for the workstation production line. A product provider was configured, a vector corresponding to the number of parts that the system will supply. The server's behavior, according to the serviced units is linear in function of time, ensuring a smooth transition in this workstation, including that at certain times the workstation remains idle, with no products to deliver to the next workstation.

In **Figure 6** shows the mean waiting time for each activity at a workstation, it can be observed that at the beginning, the server has a transient response in the products waiting time, after 40 seconds it reaches a permanent regimen that ensures a continuous waiting time. The graph shows a concordance with the behavior of the number of units served by the server's unit time, and allows one to forecast the time needed for the supply of parts to arrive at the production line, thus avoiding bottlenecks at this workstation, by holding a mean waiting time of 3 sec., this is important to predict the amount of products that this workstation can produce.



In **Figure 7** shows the server's usage time, after 40 seconds the server reaches a stationary state keeping it occupied for the whole time, thus making it possible for the calculation of the system's efficiency to supply the parts to the production line.



## 6. CONCLUSIONS AND FUTURE PERSPECTIVES

There are several formalisms used for modeling discrete event systems, but according to the literature review, the DEVS formalism is widely used today to represent this kind of systems, it is by its uses that an equivalent representation can be performed in other formalisms.

By using the Simevents® simulation tool by MATLAB® it is possible to perform the system behavior analysis that respond to discrete events, and the gathering of information regarding waiting times, service times, number of units served by a workstation, among other, which is vital, when planning an industrial production system.

The simulation was performed based on three workstations, establishing the average service time, the units produced and the setting time for each server. These results allowed the analysis and planning of load distribution in an industrial production system.

By using the DVES formalism for the modeling, and employing tools such as the MatLab® Simulink®, it is possible not only to implement simulation platforms but also the control systems, applicable in industrial production systems that answer to discrete event systems.

According to market requirements, it is necessary to structure platforms that allow the implementation of production system model in embedded processors, with the purpose of creating a system that realistically mimics the physical system, thus it may deepen the analysis of industrial production systems.

## REFERENCES

- Alur, R.; Henzinger, T.A.; Lafferriere, G.; Pappas, G.J. (2000). Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7), pp.971-984.
- Bergero, F., Kofman, E., Basabilbaso, C., Zúccolo, J. (2000) Desarrollo de un simulador de sistemas híbridos en tiempo real. XXI Congreso Argentino de Control Automático.
- Branicky, M. (1995). Studies in Hybrid Systems: Modeling, Analysis, and Control. Massachusetts Institute of Technology.
- Capocchi, L.; Bernardi, F.; Federici, D.; Bisgambiglia, P.-A. (2006). BFS-DEVS: A general DEVS-based formalism for behavioral fault simulation. *Simulation Modelling Practice and Theory*, 14(7), pp. 945-970.
- Chen, C., Yücesan, E. "Introduction to discrete event systems: Christos G. Cassandras and Stephane Lafortune; Kluwer Academic Publishers, Dordrecht, 1999, ISBN: 0-7923-8609-4, 2001.
- Filippi, J.-B. and Bisgambiglia, P. (2003). JDEVS: an implementation of a DEVS based formal framework for environmental modelling. *Environmental Modelling & Software*, 19(3), pp. 261-274.
- Hong, K.J. and Kim, T.G. (2006). DEVSpeCL: DEVS specification language for modeling, simulation and analysis of discrete event systems. *Information and Software Technology*, 48(4), pp. 221-234.
- Kain, S.; Schiller, F.; Dominka, S. (2008). Reuse of models in the lifecycle of production plants using HiL simulation models for diagnosis. *Industrial Electronics, ISIE 2008. IEEE International Symposium*.
- Kim, K. H.; Seong, Y.R.; Kim, T.G.; Park, K.H. (1997). Ordering of simultaneous events in distributed DEVS simulation. *Simulation Practice and Theory*, 5(3), pp. 253-268.
- Nikolaidou, M.; Dalakas, V.; Mitsi, L.; Kapos, G.-D. (2008). A SysML Profile for Classical DEVS Simulators. *Software Engineering Advances, ICSEA '08. The Third International Conference on, Sliema*, pp.445-450.
- Palaniappan, S.; Sawhney, A.; Sarjoughian, H.S. (2006). Application of the DEVS Framework in Construction Simulation. Simulation Conference, 2006. WSC 06. *Proceedings of the Winter*. pp.2077-2086.
- Saadawi, H. and Wainer, G. (2007). Modeling physical systems using finite element Cell-DEVS. *Simulation Modelling Practice and Theory*, 15(10), pp.1268-1291.
- Sikun, L.; Zhihui, X.; Tiejun, L. (2005). Distributed cooperative design method and environment for embedded system. *Computer Supported Cooperative Work in Design, 2005. Proceedings of the Ninth International Conference*.
- Van der Schaft, A. and Schumacher, H. (2000). An Introduction to Hybrid Dynamical Systems. *Lecture Notes in Control and Information Sciences*, 251.
- Villani, E.; Miyagi, P.; Valette, R. (2007). Modelling and Analysis of Hybrid Supervisory Systems. *Advances in Industrial Control*.
- Zeigler, B. P.; Moon, Y.; Lopes, V.L; Kim, J. (1996) DEVS approximation of infiltration using genetic algorithm optimization of a fuzzy system. *Mathematical and Computer Modelling*, 23(11-12), pp. 215-228.

**TO REFERENCE THIS ARTICLE /  
PARA CITAR ESTE ARTÍCULO /  
PARA CITAR ESTE ARTIGO /**

Amaya-Hurtado, D.; Castillo, R.; Avilés, O.; Ramos, O.L. (2014). Production System Modeling and Simulation Using DEVS Formalism. *Revista EIA*, 11(22) July-December, pp. 13-20. [Online]. Available on: <http://dx.doi.org/10.14508/reia.2014.11.22.13-21>