VALIDATING THE BEHAVIOR OF A SUPERVISED SYSTEM USING PETRI NETS

VALIDACIÓN DEL COMPORTAMIENTO DE UN SISTEMA SUPERVISADO MEDIANTE REDES DE PETRI

CARLOS ARTURO PARRA ORTEGA

Ph.D., Universidad de Pamplona, Colombia, carapa@unipamplona.edu.co

JAIME ALBERTO GUZMÁN LUNA

Ph.D., Universidad Nacional de Colombia, Medellín Campus, jaguzman@unal.edu.co

Recibido para revisar septiembre 27 de 2011, aceptado diciembre 15 de 2011, versión final enero 19 de 2012

ABSTRACT: A fundamental issue of production systems is the validating of their output. In order to obtain this output, mathematical models such as Petri nets are used to validate concurrent behaviors, presence/absence of blockings, and activity synchronization, among other aspects of this industrial process. Despite its advantages, the use of Petri nets does not allow for the evaluation of other important issues in manufacturing processes, or how to manage them. Therefore, stochastic system simulation and agent technology are used in the experiments to obtain other performance measures. A combination of Petri nets, multi-agent systems, and stochastic systems is used within a proposed method for validating the output of production systems. This paper shows how this method is used in an academic manufacture sample.

KEYWORDS: manufacturing systems, Petri nets, stochastic simulation, DEVS, logic agents.

RESUMEN: Un aspecto fundamental de los sistemas de producción es la validación de su comportamiento. Paraobtener este comportamiento, se recurre frecuentemente a modelos matemáticos tales como redes de Petri, con el fin de validar su comportamiento concurrente, presencia/ausencia de bloqueos, y sincronización de actividades, entre otros aspectos. A pesar de sus ventajas, el uso de redes de Petri no permite evaluar otros aspectos que son importantes en los procesos de manufactura, así como su gestión. Por tal motivo se recurre a la simulación de sistemas estocásticos y a la tecnología de agentes para obtener otros índices de desempeño. En este artículo se propone una combinación de redes de Petri, sistemas multi-agentes y sistemas estocásticos en un método propuesto para validar el comportamiento de los sistemas de producción, y se presenta como este método es usado en un ejemplo académico de manufactura.

PALABRAS CLAVE: sistemas de manufactura, redes de Petri, simulación estocástica, DEVS, lógica de agentes.

1. INTRODUCTION

The purpose of this document is to present a method for validating the supervision of an industrial process. This method combines Petri nets, multi-agent systems, and stochastic systems to simulate and jointly evaluate the behavior of the subsystems of control, supervision, and management of an industrial organization. The structure of this document is as follows: In Section two we will describe a methodology to evaluate the dynamical behavior of supervised systems. In Section three, a study case is presented to apply the aforesaid method; in Section four, we will describe the physical and logical model of the example. Section five shows the synthesis process and implementation of a supervisor system, while the validation by multi-agent simulation and discrete-event that show the behavior of the process are described in Section six. Also, results are commented upon there. Finally, conclusions are presented.

2. VALIDATION METHODOLOGY OF A SUPERVISORY CONTROL SYSTEM

To design supervisory control systems, it is essential to have a procedure that allows the determination of the validity of a supervisor's synthesis and its implementation afterwards. This procedure includes synthesis aspects of supervisory systems, from system simulation to discrete events and multi-agent systems [7, 11], and an application of a specific case depicted in

[2]. We can schematize in Fig. 1 the idea of validating the behavior of these systems given their behavior specification. As you can see there, there are three levels for the application of a supervisory control system.

At a level close to the plant floor, we have an industrial process whose most representative variables are continuously measured by means of sensors and controllers, which leave a record of the measurements taken. The supervisor uses this data to produce, later on, commands that it sent to the controllers to assure the trajectory of the variables into a specific range of values. As a result of the difficulty in experimenting with physical industrial systems, it is necessary to have validation of the dynamics of their behavior using modeling and simulation. Hence, an industrial process is projected toward a hybrid model that combines discrete and continuous variables simulating a measuring process and assigning supervisor's functions to behavior models based on a logic of agents.

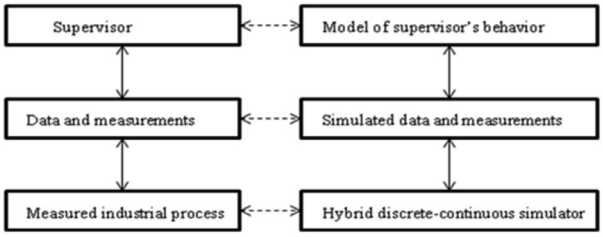

Figure 1. General scheme for validating the behavior of a supervised process

Due the complexity of the industrial process that we wish to supervise, the methodology proposed is of the iterative type in which the starting point is formed by a description of the physical process to control, and reference architecture of industrial organization. Reference architecture determines the way activities will be coordinated within the process in such a way that it is not the same to coordinate a set of processes hierarchically in which planning and supervision are centralized than to coordinate a holarchy of production units in which each one plans and supervises its own process, coordinating with each other via negotiation or commands.

In Fig. 1 it is possible to see an iterative process. Notice that once you have a physical model and organizational architecture, you move on to defining the logic of the process, establishing operating conditions as a discrete state system, as well as having an availability of resources, products, and production methods. Reference architecture has an influence on the size of a discrete system, since a system resulting from the composition of various subsystems has greater state space.

We can find similar work in [9] where Leitao describes the behavior of a holonic system by means of Petri networks, and evaluates via indicators such as throughput and resource use, among others.

3. DESCRIPTION OF THE CASE STUDY

The problem of hydropneumatic system control is presented as an example to apply supervisor's control, modeled on discrete event systems and multi-agent system simulation. The system supervised includes a cylindrical water reservoir which has air and water. Water is an incompressible fluid, and its level is being controlled by the supervisory control system so that it may remain between two values, minimum and maximum. Air is compressible, and as water enters the water reservoir, pressure increases and it determines the quality of a water supply for users' demand of water. This pressure must also be controlled in such a way that it varies between values' minimum and maximum. If pressure decreases beyond the minimum value, air is injected using a compressor. If the water level decreases beyond the minimum value, water must be pumped using one or two motorpumps and valves that allow water reservoir input flow; while if it increases beyond the maximum value, pumping stops and the input valves close. The schema representing this system may be seen in Fig. 2. The objective of the system includes supplying water to users, keeping its reservoir levels and air pressure within an adequate rank of values. To do so, a new system of agents will be proposed to implement a supervisory control system, and the dynamic evolution of this system will be validated by means of modeling and simulation, as we will describe later on. It is considered that the plant or system to be controlled is the environment that surrounds the agents in charge of managing each production unit.

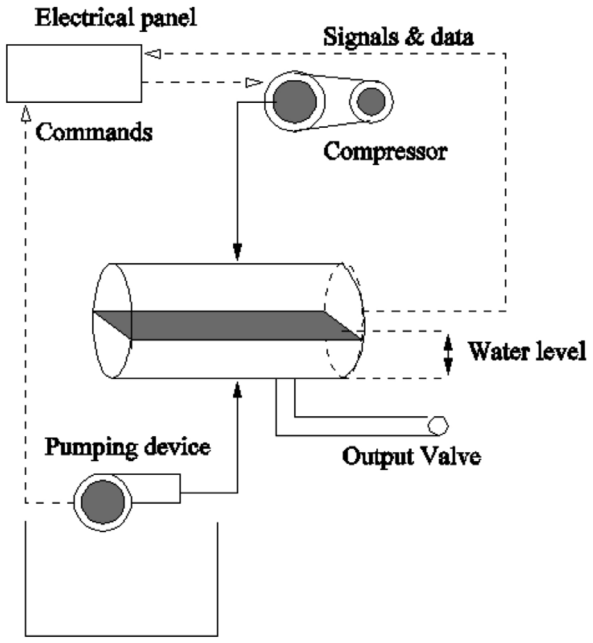

Figure 2. Scheme of a hydro-pneumatic system

The design conditions that must be fulfilled are as follows:

- The reservoir is cylindrical, and its radius and height are known.
- Rising pressure must be between 80 and 100 lbs. per square inch.
- Only one pump must work at 95 lbs. of pressure.
- The reservoir level must fluctuate between 20 and 60% of the maximum water level allowed in the tank.
- Pumps must not work in a vacuum. A minimum level of water is required in the tank
- During the filling process, an external demand of water may arise, so that there is an output flow.
- The input valve must be closed when the reserve tank reaches a predetermined level.

4. PHYSICAL AND LOGICAL MODEL OF THE SAMPLE PROCESS

4.1. Physical model in a continuous variable

Bear in mind that there are two reservoirs; one of them has the shape of a horizontal cylinder with an r radius. Volume V of the reservoir is given by the expression $A_c h$, where h is the height of the liquid level and A_c is the area of the circular segment whose height is h. As a control policy, it is considered that the reservoir cannot contain more than a percentage of its total volumetric capacity to avoid overflowing, so that there is a maximum height of the water level for each tank, represented by the variable h_{max}, and a minimum permitted height which is denoted with the variable h_{min}. The differential equation that relates water level with input and output flows is

$$h'(t) = \begin{cases} \dfrac{\lambda(t) - \mu(h(t))}{r^2 \cos^{-1}\left(\dfrac{r-h}{r}\right) - (r-h)\sqrt{2rh-h^2}} \end{cases} \quad (1)$$

Where $h'(t)$ is the change rate of the water level in the tank and h is an instantaneous height, measured at a determined moment. Due to the physical conditions and the control of the system, the value for variable $h(t)$ will be restricted by means of the following inequalities: $h_{min} \leq h(t) \leq h_{max}$, for any instant of time t posterior to 0, since at that time the tank starts its filling process. To comply with this restriction, input flow is regulated by means of a valve, and output depends on demand. Input and output flows are described by the following expressions:

$$\lambda(t) = \begin{cases} k, \text{ if } h(t) \leq h_{max} \\ 0, \text{ opposite case} \end{cases} \quad (2)$$
$$\mu(t) = \text{constant C}$$

The interpretation given to these two equations is that the input flow of one of the tanks may be considered constant while the water level in the tank does not exceed h_{max}, and remains on 0 while its level does not drop under h_{min}.

The other reservoir is external, and it is cylindrical with a radius R, where one assumes that its level and its input and output flows are controlled externally.

These two reservoirs are connected by a duct through which a pumped fluid flows from the external reservoir to the internal reservoir in such a way that the output flow of the external tank will be the input flow of the other tank after a time t_a. Since the amount of water pumped must pass from the external tank to the internal tank within the duct, t_a indicates the time required in order to complete the trajectory. Regarding pressure control, you start from the fact that air is a compressible fluid; therefore, a determined volume of air may be confined to a lower volume. This confinement has a

physical effect that increases resulting pressure and its behavior may approximated with the following equation, supposing that air will behave as an ideal gas:

$$\frac{P_0 V_0}{T_0} = \frac{P_1 V_1}{T_1} \quad (3)$$

where P_0 is the pressure when the volume occupied is V_0. Supposing that this process is isothermal, temperatures T_0 and T_1 are equal, so it may occur that if the available volume is reduced for the air; then its pressure will increase and vice versa.

4.2. Operating conditions

To make the projection of the values of continuous variables easier, it is necessary to establish a set of discrete states; the following operating regions have been defined and listed below:

- Rest. This happens when the water level is zero (empty), the pump is off, and the valve is closed.
- Filling preparation phase. The water level is still zero, the pump is shut down, and the water input is still zero.
- Water level is still zero, the pump is shut down, and the external input valve is open.
- Minimum filling phase. The water level is below 1.2 m, the pump is on, the input valve is open.
- Filling phase. The water level is between 1.2 and 4.6 m, the pump is on, the input valve is open.
- Full-filling phase. The water level is over 4.6 m, the pump is on, the input valve is open.
- Draining preparation phase. The water level is over 4.6 m, the pump is off, the input valve is open.
- Minimum draining phase. The water level is over 4.6 m, the pump is off, and the input valve is closed.
- Draining phase. The water level is between 1.2 and 4.6 m, the pump is off, and the input valve is closed.
- Full draining phase. The water level is over 1.2 m, the pump is off, and the input valve is closed.

4.3. Process logic modeling as a discrete event system

To execute a process supervision application, it is necessary to know a model of its discrete dynamics. The presence of operating conditions and states suggests that the process has a discrete nature even if there are continuous variables to control. To obtain a process model with this characteristic, it is possible to describe it by using two approaches: utilizing finite-state machines or Petri nets. There are many reasons to describe discrete systems using Petri nets since they can provide information regarding the structure and behavior of a system; moreover, this information can be used to evaluate a modeled system and suggests improvements or changes. Another quality is that Petri nets describe synchronic or asynchronic concurrences. As a result of the operating properties of this process, the Petri net approach was chosen since it determines the state of the system in accordance with the position of messages (tokens) on a graph, and its performance can be simulated as a set of objects exchanging messages [5]. Furthermore, it simplifies the design of a supervised controller without having to significantly increase network complexity.

A formal definition of a Petri net establishes that it has four elements: a set of places *P*, a set of transitions T, an input function I, and an output function O, which relate transitions and places. This formal definition was established by Peterson [13], which sets forth that a Petri net is defined as a C quadruple, in which C = (P,T,I,O).

Where

- $P = \{p_0, p_1, p_2, \ldots, p_n\}$ set of places, $n>0$
- $T = \{t_0, t_1, t_2, \ldots, t_m\}$ set of transitions, $m>0$
- $I: T \rightarrow P\infty$ is the input function that projects from transitions to places
- $O: P \rightarrow T^{\infty}$ a mapping from transitions to groups of places

It is worth highlighting that there are other formal definitions equivalent to the aforementioned, in which they define a Petri net from a matrix standpoint in accordance with Moody's work [10], destined at the synthesis of process controllers. The description of the dynamics of a

process represented by Petri nets is based on the concepts of events and conditions. Events imply changes of state within the system and their occurrence is instantaneous in time; conditions are logical descriptions of the state of various parts of the system. So that events may take place, certain conditions must be fulfilled, called *preconditions*. When an event takes place, these pre-conditions generally change to create another set of conditions called *post-conditions*. Thus, the post-conditions of an event may be the preconditions of anther event. To construct a Petri net, the modeler must estimate the interpretation domain of the physical system, labeling places and transitions. Hence, by identifying system events and conditions, one can model a system of discrete events as follows:

- Every event is presented and labeled as a transition.
- Places represent conditions.
- The preconditions of an event are the input points of a transition of that event.
- The postconditions of an event are the output points of a transition that represents an event.

The Petri net that represents the production process being analyzed is shown in Fig. 3. Every state in which you can find a reservoir is linked with a place, just like ones on the input valve and on the pumping device. The meaning of each place and transition will be described later on.

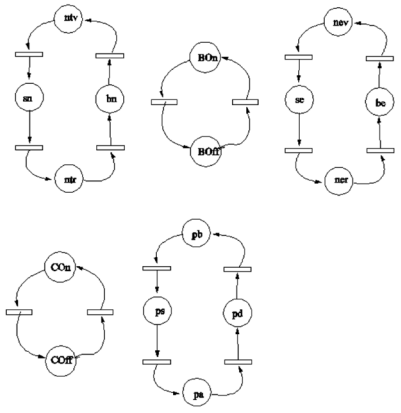

Figure 3. Petri nets representing the process

4.1.1. Places and transitions linked to the process

Places are linked to the state of the reservoir, valves, or pump; and transitions indicate the occurrence of events that modify a state or tasks carried out to attain a determined state. The following tables describe each one of them, respectively:

Table 1. Description of the places on the Petri net of the case study

Name	Description
NTV	Empty main reservoir or level below minimum
SN	Ascending main reservoir level
SB	Descending main reservoir level
NTR	Main reservoir with a level surpassing maximum
BOn	Pump on
BOff	Pump off
NEV	External reservoir with a level below minimum
SE	Ascending external reservoir level
BE	Descending external reservoir
NER	External reservoir with a level over the maximum
PB	Low internal reservoir pressure
PS	Ascending internal reservoir pressure
PD	Descending internal reservoir pressure
PA	High internal reservoir pressure
COn	Compressor on
COff	Compressor off

Table 2. Description of Petri net transitions which depicts the discrete system

Name	Description
T1	Changes net token from place NTV to SN, internal reservoir surpasses the minimum level
T2	Changes net token from place SB to NTV, descending internal reservoir goes beyond the minimum level
T3	Changes place from SN to NTR, reservoir level exceeds maximum
T4	Changes place from NTR to SB, internal reservoir level descends from maximum
T5	Changes place from BOn to BOff, turn off pump
T6	Changes place from BOff to BOn, turn on pump
T7	Changes place from NEV to SE, external tank starts filling
T8	Changes place from BE to NEV, external tank level descends beyond minimum
T9	Changes place from SE to NER, external tank level surpasses the permitted maximum
T10	Changes place from NER to SB, external tank level descends below maximum
T11	Compressor is shut off, going from place COn to COff
T12	Compressor is turned on, going from place COff to COn
T13	Increased pressure going from place PB to PS
T14	Pressure surpasses maximum level,going from place PS to PA
T15	Descending Pressure, going from place PA to PD
T16	Reservoir air pressure descends to minimum, going from place PD to PB

In this example, some places on the Petri net depend on the evolution of continuous variables, so that we define a hybrid Petri net just like transitions from one place to another; particularly, places and transitions which correspond to the state of the levels of the reservoirs and pressure.

5. SYNTHESIS OF SUPERVISOR SYSTEM

As Moody and Antsaklis [10] propose, the most direct way to synthesize a supervisor consists of applying lineal restrictions. But for the specific case of the hydropneumatic system, events that change the state of levels and pressure are non-controllable. Hence, this method of synthesis cannot be applied. Another way to obtain a supervisory control system consists of relying on the theory of languages of events, where a language of events is a sequence of events. A supervisor may react according to a determined sequence to induce events that produce a new sequence, or maintain a trajectory of events.

5.1. Heuristic synthesis based on languages of events

To obtain a supervisory control system that allows you to avoid conditions that lead to an unwanted state, you have to introduce the following restrictions: every time that a transition 3 is triggered, indicating that the internal reservoir has reached the state of overflow, there is a triggering of transition 5 by means of control *C1*, which opens the suction pump and vice versa with transitions 2 and 6, which joins in to shut off the pump by means of place *C2*. The same thing takes place when transition 15 is triggered, which indicates that the pressure in the external reservoir has increased beyond the permitted maximum. This led to the enabling of transition t11, which shuts down the compressor just like transitions 14 and 12 turn it on. The system of the external tank is not under control, but knowing its internal state allows one to take actions in order to not pump when its level decreases to minimum. To do so, one resorts to an inhibitory arc to impede the triggering of the suction pump from place *C5*, activated by transition 8; when the level returns to normal, place *C6* is triggered by transition 7, lifting the restriction regarding turning on the pump. For simulation purposes, which will validate this system of discrete events, consider the conditions for triggering a transition in which the token is in the place that enables it and some rules exist to classify the qualitative state of the variable that describes liquid level *h(t)* and pressure *p(t)*. In Fig. 4, we show how the discrete system is totally integrated, and the actions of supervisory control system have been linked to the discrete systems the resources stand for and the state of the process.

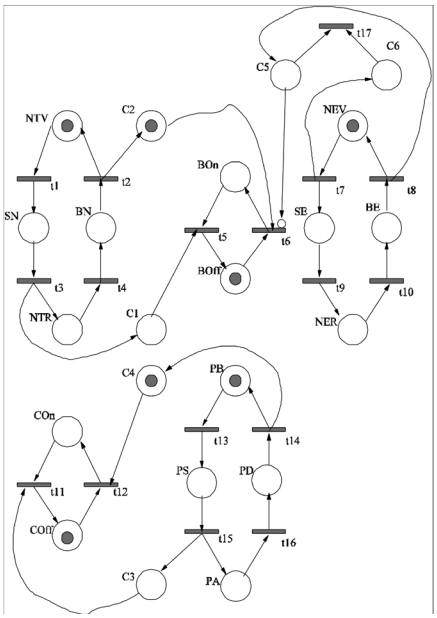

Figure 4. Petri net representing the entire supervised system

In Table 3, one can see the actions of a supervisory control system.

Table 3. Actions of a supervisory control system in a process

Name	Description
C1	Action of supervisory control system to shut down the motorpump when the maximum level of the horizontal tank is surpassed
C2	Action to turn on the motorpump when the water level goes under the minimum level allowed
C3	Action to shut down the compressor
C4	Action to turn on the compressor
C5	Action to prevent turning on the motorpump if the external tank is empty
C6	Action to lift the warning to not turn on the pump
C7	Action to prevent the pump from turning on if pressure is high
C8	Action to lift the warning to not turn on the pump because of pressure

5.2. Specification of a Supervisor's Behavior

To model a supervisor's rational distributive behavior, we resort to agent technology. To specify a supervisor's behavior that we will validate through simulation, we will resort to the architecture of reasoning proposed by Kowalski and others [8] where they state that an agent combines two types of reasoning: reactive and rational, which allow it to react when there are environmental changes and design action plans. In our particular case,

we have identified three roles that may be implemented by agents: an event-detecting agent, a supervising agent, and an actuating agent. Below, you can see a specification fragment of an event detector, using for them a sublanguage called *actilog* to simulate reactivity [3], which is implemented by a logic programming language called Prolog.

If read,reading-level(Z,X),Z==1,X==0,valve(W,Y),W==1,Y==’off’ then rep-empty-te.

If read,reading-level(Z,X),Z (Z,X),Z==1,X==0,valve (W,Y),W==1,Y==’on’ then rep-ok-te.

If read,reading-level(Z,X),Z==1,X < 1.2,valve(W,Y), W==1,Y==’off’,operating then rep-low-te.

If read,reading-level(Z,X),Z==1,X < 1.2,valve(W,Y), W==1,Y==’on’,operating then rep-ok-te.

If read,reading-level(Z,X),Z==1,X > 1.2,X < 2.9, operating then rep-ok-te.

If read,reading-level(Z,X),Z==1,X > 2.9,valve(W,Y),W==1,Y==’off’, operating then rep-ok-te.

where read,reading-level *(A,B)* are observations the agent carries out; while rep-low-ta is an action which leads to have the detector report that the level of the external tank has decreased.

6. SIMULATION TO EVALUATE SYSTEM PERFORMANCE

Petri nets are tools to validate the behavior of a model regarding the synchronization of its activities and the design of answers in view of the occurrence of events, but it does not establish a pattern regarding how the system will behave in time, for instance, when each event will occur. Thus, it is difficult to establish a measure of efficiency in the system. For this case study, a question that could be asked is what percentage of time is the system in a determined state, or how much is the excess liquid while you detect an overflowing event before closing the valve. Replying to this question requires conducting discrete event simulation experiments. System simulation of discrete events by means of a list of future events [1,12] or by process interaction [6] offers a complement to measure system efficiency regarding the degree of equipment use, if there is or if there is not a pooling of inputs, a probability of occurrence of an event in a given time, or other aspects.

Models that represent discrete event systems are dynamic, stochastic, and discrete; where state variables change the value at unpredictable times at given instants. These instants in time correspond to an event; therefore, “an event is defined as an instantaneous action that may change the state of the system” [6]. Summarizing various authors, the elements of interest of discrete event models are:

- Activities. These are tasks that are carried out in the system, delimited by two events: one that starts it and one that ends it.
- Entities. These are objects that circulate through the system. They may be (permanent) resources the system uses or objects processed in the system (temporary).
- Attributes. These are the properties of resources and of the temporary objects of the system.

6.1. Definition of the model as net of components

To simulate the system that contains continuous dynamics represented by equations in Section 1 with the discrete dynamics shown in the previous section, a mathematical model was implemented in the GALATEA simulation platform. This simulation platform implements the specification of a simulation language aimed at discrete events and with software agent technology. Figure 5 shows a net of nodes that represents both discrete and continuous processes, and it also shows the flow of the messages that will go throughout the net carrying data. A description of the meaning of each node can be observed in the GLIDER [4] reference manual. For our case, the places on the Petri net are *Gate* type nodes; in other words, they hold messages until a transition occurs. Transitions correspond to generic type nodes, and they activate periodically by means of an autonomous node called a *sensor*, which simulates a reading measured from the *continuous* node that represents the liquid reservoir. The *supervisor agent* manages the entire system interacting with agents as a detector and an actuator, which interact with the system (environment) via an interface, where they observe the events that have taken place, and they induce new events by means of

an actuator. These events may be a) opening an input valve, once enabled to do so; b) closing an input valve, once the event has been enabled; as well as, starting and shutting down the suction pump. In our case, the entities that are going to travel through the net of components are messages that indicate the marking in effect in each place (gate node) on the Petri net. The only component which is triggered periodically is the supervisor, which activates other agents when required, just like a real sensor. All other components are only activated when there is an event whose consequence is that activation.

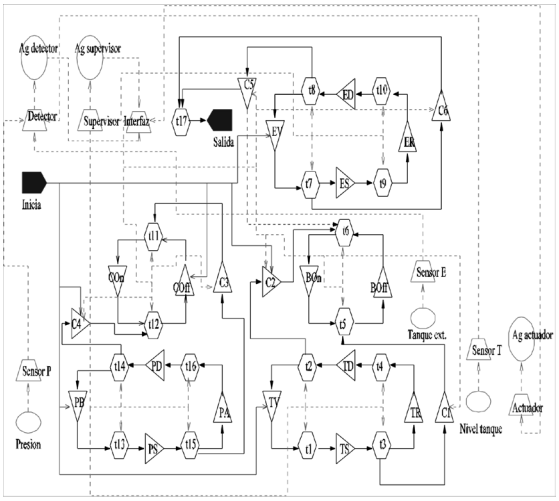

Figure 5. Scheme of queue nets representing a discrete and continuous system

6.2. Initial conditions

To execute a simulation, we establish the initial conditions of the system, where the physical conditions are as follows:

The height of the level of the horizontal tank is 0.4 m. The pressure inside the horizontal tank is 14.5 lb-ft^2. The height of the level of the external tank is 2.8 m. The output flow by default is 0.01 m^3/sec. Compressor capacity is 2 dm^3/sec.

The initial marking of a Petri net which represents every industrial process, plus a supervisor's action is

$M = [1\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0]^T$

Where the first four elements of the vector correspond to the states of the level of the horizontal tank, the following two describe the state of the motorpump, and the last 6 elements are linked to the supervisor's direct action.

6.3. Analysis of results

After executing the DEVS model with the initial conditions previously established, and given the specification of the agents' behavior, there were three ways to analyze this simulation: a trace of continuous values of variables (such as time, level and pressure), the marking graph of the Petri net process, and a control trace of the commands sent to the supervisory control system.

Moreover, it is also possible to obtain a trace of observations which the agents of their environment make such observations and nominations of actions. An example of this trace for the event detector is presented below:

observed [operating,read,reading-level(1,2.8), reading-level (2,0.3332771485576115), valve(1,off), valve(2,off), reading- pressure(18.719137546804983), compressor(off), time(10.0),null] my influences are: [rep-ok-te, rep-low-ta, rep-ok-pr]

```
observed [operating,read,reading-
level(1,2.8), reading-level
(2,0.3300055), valve (1,off), valve
(2,off), reading- pressure(18.6154525)
,compressor(off), time (20.0),null] my
influences are: [rep-ok-te]
```

The same happens with other agents that intervene in the system.

7. CONCLUSIONS

Two types of conclusions can be drawn when carrying out the experiment of evaluating the behavior of supervised production units.

A combination of methods were tested to evaluate the supervisory control in continuous production systems, combining aspects of hybrid systems, projecting to discrete event systems a synthesis of supervisors based on a language of events, multi-agent systems, and DEVS components.

You could automatically generate the behavior rules of an agent which implements a supervisor, observing pre and post conditions of the triggering of transitions on a Petri net which models the behavior of the entire supervised process, and then establishing the rules for triggering the supervisor's transitions, as long as they are controllable.

The supervision of a production unit can also be represented through finite state automats, which are based on the language of the sequence of events. The DEVS paradigm simulates a complete industrial process with both resources and stages of the process within one same model combining stochastic queues representing the use of resources, which are described by continuous variables with concurrent models based on an activity like Petri nets.

As future work, we recommend simulating the coordination of various production units the same way to evaluate how much influence negotiation processes have, and the composition of the states of each production unit in an enterprise.

ACKNOWLEDGEMENTS

The authors thank the research group in Computer Science (CICOM) at the University of Pamplona for kindly providing many of the main ideas of this analysis. The project was partial sponsored with resources from DIME project #20201009532, "Programa de Fortalecimiento del Grupo de Investigación Sistemas Inteligentes – SINTELWEB - Convocatoria Nacional 2010 -2012, Modalidad 3" of Universidad Nacional de Colombia.

REFERENCES

[1] Banks, J.; Carson, J.; Nelson, D.; Nicol, E. Discrete-Event System Simulation. Prentice Hall International Series in Industrial and Systems Engineering, 1999.

[2] Chacon, E.; Parra, C. Evaluation of the performance of manufacturing systems projecting petri nets towards discrete event nets. XII Latin American congress in automation. CLCA, Brazil, 2006.

[3] Davila, J. Agents in logic programming. Ph. D. thesis, Department of Computing, Imperial College. London University, 1997.

[4] Domingo, C.; Tonella, G.; Sananes, M. GLIDER's reference manual. Universidad de Los Andes, Mérida, Venezuela, 1996.

[5] Fishwick, P. Simulation model design and execution. Building digital worlds. Prentice Hall International Series in Industrial and Systems Engineering, 1995.

[6] Guasch, A.; Piera, M.; Casanovas, J.; Figueras, J. Modeling and simulation. application of logistic processes, manufacturing processes and services. Modelado y Simulación. Aplicación a procesos logísticos, fabricación y servicios. Editorial Alfaomega, 2005.

[7] Jiménez, J.; Ovalle, D.; Ochoa, F. SMART: Multi-Agent Robotic System. Dyna-Colombia Vol. 154, pp. 179-186, 2008.

[8] Kowalski, R.; Sadri, F. Towards a unified agent architecture that combine rationality with reactivity. LID'96 workshop on Logic in databases, 1996.

[9] Leitao, P. An agile and adaptive holonic architecture for manufacturing control. Ph. D. Thesis. University of Porto, 2004.

[10] Moody, J.; Antsaklis, P. Supervisory control of discrete event systems using Petri nets. Kluwer Academic Publishers, 1998.

[11] Parra, C. Modeling and simulation of supervisory control for continuous production holonic systems. Modelado y simulación del control supervisorio para sistemas holónicos de producción continua. Ph. D. thesis. Universidad de Los Andes, Venezuela, 2009.

[12] Law A.; Kelton, D. Simulation modeling and analysis. McGraw-Hill Series in Industrial Engineering and Management Science, 2000.

[13] Peterson J. Petri net theory and the modeling of systems. Englewood Cliffs: Prentice-Hall International, 1981.