Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

MODELING AND DEVELOPMENT OF AN ARENA® INTERFACE FOR PETRI NETS. A CASE STUDY IN A COLOMBIAN COSMETICS COMPANY

Gonzalo Mejía
Dept. of Industrial Engineering
Universidad de los Andes
Bogotá, Colombia

Diego Martínez
Mimesis Soluciones Ltd.
Bogotá, Colombia

Fidel Torres
Dept. of Industrial Engineering
Universidad de los Andes
Bogotá, Colombia

ABSTRACT

This paper presents an ARENA®-based library for the development of Timed Colored Petri Net (TCPN) models. In this way, the strengths of Petri Nets and Discrete Event Simulation (DES) are combined in a powerful tool for modeling and analysis of systems. The use of the library is illustrated on a cosmetics manufacturing facility in Bogotá (Colombia). This facility was modeled via both TCPNs and a classical DES methodology. Extensive analysis shows the advantages and disadvantages of each approach.

1 INTRODUCTION

Petri Nets have been widely used for modeling manufacturing systems. Petri Nets can handle parallel machines, buffers of finite capacity, dual resources (multiple resources required simultaneously on one operation), alternative routings, and material handling devices to name a few (Dubois and Stecke 1983). In addition, Petri Nets provide mathematical tools for simulation, analysis and validation that can help detect deadlocks or the feasibility of a production plan (Mejía ands Odrey 2005).

Despite its popularity, most Petri Net software has been developed for research and academic purposes and therefore the possibilities of technical support, comprehensive testing, and training are limited. A feasible solution for this problem is the development of external plug-ins or libraries that would interact with commercial DES software packages.

This paper presents HPNM (Hybrid Petri Net Model) which is an ARENA®-based library for the development of Timed Colored Petri Net (TCPN) models. This library uses the ARENA® tools and defines the entities, attributes and entity flow required to fully describe the functioning of a TCPN. A user-friendly interface helps the model development.

A main objective of the present paper is to analyze and compare two types of simulation. The first alternative is DES and the second is a HPNM running on the ARENA® simulation software. The comparisons are illustrated on a cosmetics manufacturing facility in Bogotá (Colombia). This facility was modeled via both TCPNs and a classical DES methodology. The paper concludes by illustrating the advantages and disadvantages of each approach.

2 BACKGROUND

2.1 Petri Nets

A Petri Net is a particular kind of directed graph, together with an initial state called the initial marking. A Petri Net is a directed, weighted, bipartite graph consisting of two kinds of nodes, called places and transitions which respectively represent conditions and events (Desrochers and Al-Jaar 1995)

Additional entities denoted as tokens reside in the places and represent the truth of the condition associated with the corresponding place. A marking is defined as an array which contains in each of its positions the number of tokens at each place. Tokens move throughout the net by effect of transition firings. If a transition fires, one token is removed from all the transition input places and one token is put on each of the transition output places. Before a transition fires, it must be enabled. This occurs when all its input places are marked (contain at least one token). A marking (state of the net) M' is reachable from another marking M if there exists a valid sequence of transition firings which converts M into M'. A Timed Petri Net (TPN) considers time associated with either places or transitions. In this paper was considered time associated with transitions. In TPNs, in addition to the token rule for transition firing, a transition is enabled when the time associated with the transition has been exhausted.

The above TPN representation usually implies a large number of places and transitions when modeling real systems. If the structure of an ordinary Petri Net has several similar constructs then the net can be simplified with the use of "colors" or attributes. The term "color" was coined by Kurt Jensen who created the Colored Petri Nets

(CPNs) (Jensen 1981). These colored nets combine the power of ordinary Petri Nets with the potential of high-level programming. As such, the ordinary PN provide the operation or interaction, while the programming language brings the possibility of definition of data types and manipulation of variables. This representation is the basis of the different behavior properties and its analysis.

CPNs as in the case of their counterpart PNs, have a formal mathematical representation as well as an extensive study and theory. However, for the practical uses of the CPN and their tools, is enough to have an understanding of how they function. CPNs can give explicit reference to the time or not. The CPNs without time are used to validate the functionality of a network, while the temporized CPN (TCPN) are used to evaluate or simulate a system.

For example, consider a manufacturing system in which all jobs follow the same route: Instead of defining a route for each job, a Colored Petri Net has a common route (net) for all jobs and such jobs are differentiated by special attributes denoted as colors (there is a color for each job). Specific rules must be defined for the transition firings. This establishes which colored tokens must be present at the input places before the transition firing and which colored tokens result at the output places when the transition fires. This set of rules is denoted here as the Color Function and it is defined for each transition (Jensen 1981).

Definition. Timed Colored Transition Petri Net (TCPN): A TPN is a 6-tuple $(P, T, P \times T, T \times P, C, M_0, \tau)$ where P= set of places, T = set of transitions, $P \cap T = \{\}$, $P \times T$ = a set of input arcs, $T \times P$ = a set of output arcs, C is the color function associated with each transition, M_0 = Initial marking array, and τ = set of time delays associated with transitions. An arc from place p_i to transition t_j is denoted here as $(p_i\text{-}t_j)$. Conversely the arc connecting transition t_j with place p_i to is denoted here as $(t_j - p_i)$.

This paper followed the methodology concept of S^4R nets (Systems of Sequential Systems with Shared Resources (Zhou and DiCesare 1993; Mejía and Odrey 2005) for modeling manufacturing systems. In S^4R nets, a number of concurrent serial processes (jobs) are modeled with an equal number of strongly connected state machines (nets in which each transition has only one input and one output places). Places belonging to such processes are denoted as "operational places". Resource capacity constraints are incorporated to the model via "resource places", i.e. the maximum number of tokens at any resource place represents the number of available resources of a type.

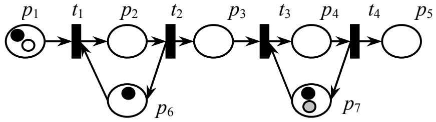

Figure 1: Colored Petri Net

Figure 1 illustrates a Colored Petri Net which represents a simple manufacturing system. Notice that two colored tokes reside at both p_1 and p_2. Tokens at p_1 represent that two distinct jobs and tokens at p_7 represent the availability of two different resources at a single workstation.

Table 1: Description of Places

ID	Condition or Action
p_1	Job ready
p_2	Job being processed by machine 1
p_3	Operation 1 of Jobs finished
p_4	Job being processed by machine 2
p_5	Operation 2 of Job 1 finished
p_6	Machines at workstation 1 available
p_7	Machines at workstation 2 available

2.2 ARENA

ARENA ® is World wide leader in DES software. In the 2005 Winter Simulation Conference about 45% of the papers showed ARENA-based applications.

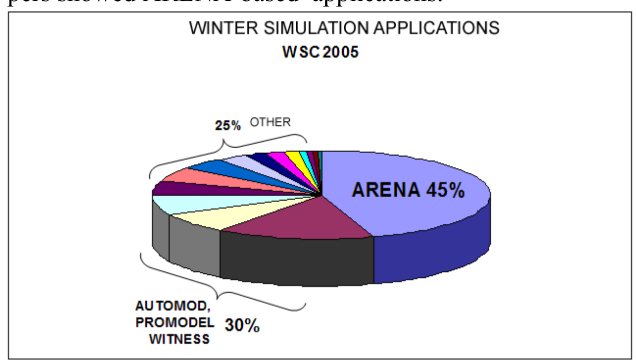

Figure 2: Application Software Distribution.

ARENA is a graphical interface whose underlying language is SIMAN (Law and Kelton 2000). SIMAN is a low level simulation language which provides the basic elements for DES. Those elements include event list management, generation of random numbers, and collection of data for statistical purposes.

The simplest modules in ARENA are elements and blocks (Kelton, Sawdowski, and Sturrock 2004). The elements represent basic language structures. The main ARENA elements are:

- Entities: These are structures that flow across modules. Each entity moves when an event is triggered.
- Resources: These are elements required to perform a task.
- Queues: These are elements in which the entities wait for the availability of resources.

- Variables: These are global data structures which can be accessed and modified by the entities.
- Attributes: These elements are local variables specific to individual entities. an attribute might be the skin color or sex. A variable might be the number of male pedestrians, the number of female pedestrians, etc.

Blocks are intended to perform actions on the elements. Typical actions are changing the values of entities and / or variables, making decisions such as picking a specific queue and controlling the flow of entities. The blocks used in this application were:

- Create: Creates new entities. In a PN, it will be used to make the initial marking.
- Branch/Decide: picks a path depending on a condition e.g. priority) or probability.
- Hold and Signal. The Hold block stops the flow of an entity until a condition is met. When such a condition is met, the Signal block issues a message to the Hold block and the entity is released. This block can be used to move the tokens from a place to the transition
- Assign: This block changes the values of variables and or attributes.
- Separate: This block is used to duplicate entities.

3 THE HPNM LIBRARY

Templates are tools for developing libraries of programming languages and perform specific tasks. In ARENA, the construction of libraries has three main parts:

- A graphical representation
- The programming logic
- The graphical interface

The graphical representation contains the ARENA logic that describe the behavior of the elements of the system; the programming logic is based in six modules, namely:

- Place
- Colored Place
- Timed Transition
- Timed Colored Transition
- Conflict Resolution
- Colors

Figure 3 shows the logic of a colored place. This place has four processes:

1. Collection of statistics: Collects statistics such as queue length, and queue waiting time.
2. Set of queues: Contains the logic for token movements. Uses Hold blocks to ensure that tokens (entities) only move when transitions are enabled.
3. Transition routing: Defines the logic that transfers tokens to the appropriate transitions with the appropriate numbers.

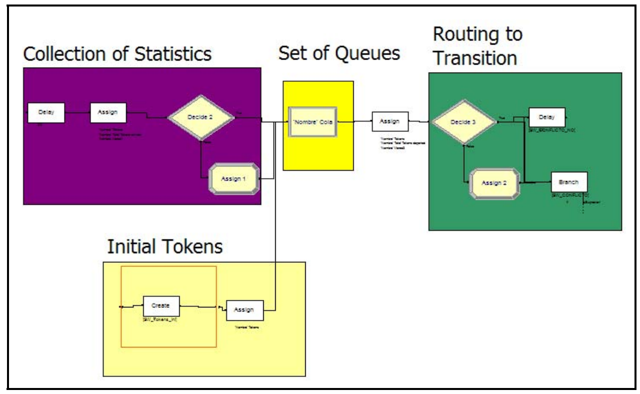

Figure 3: Colored place modeling

Figure 4 shows the logic of the module Timed Colored Transition. The transition module has six processes:

1. Firing evaluation: Establishes whether the transition is enabled with respect to a specific color.
2. Conflict Resolution: This process uses the Hold and the Search block to resolve firing conflicts (i.e. more than one transition can fire at a given time).
3. Token Removal: This process extracts tokens from the transition inoput places according to the firing rule (which color) and the input weights (how many tokens). This process uses a Signal block to send messages to the input places in such a way that the appropriate tokens are released.
4. Token Addition: This process creates the tokens that will be put at the transition output places.
5. Transition Delay: This specifies the transition processing time which can be either deterministic or stochastic.
6. Animation Update (Actualización de la Animación): This is the token game animation.

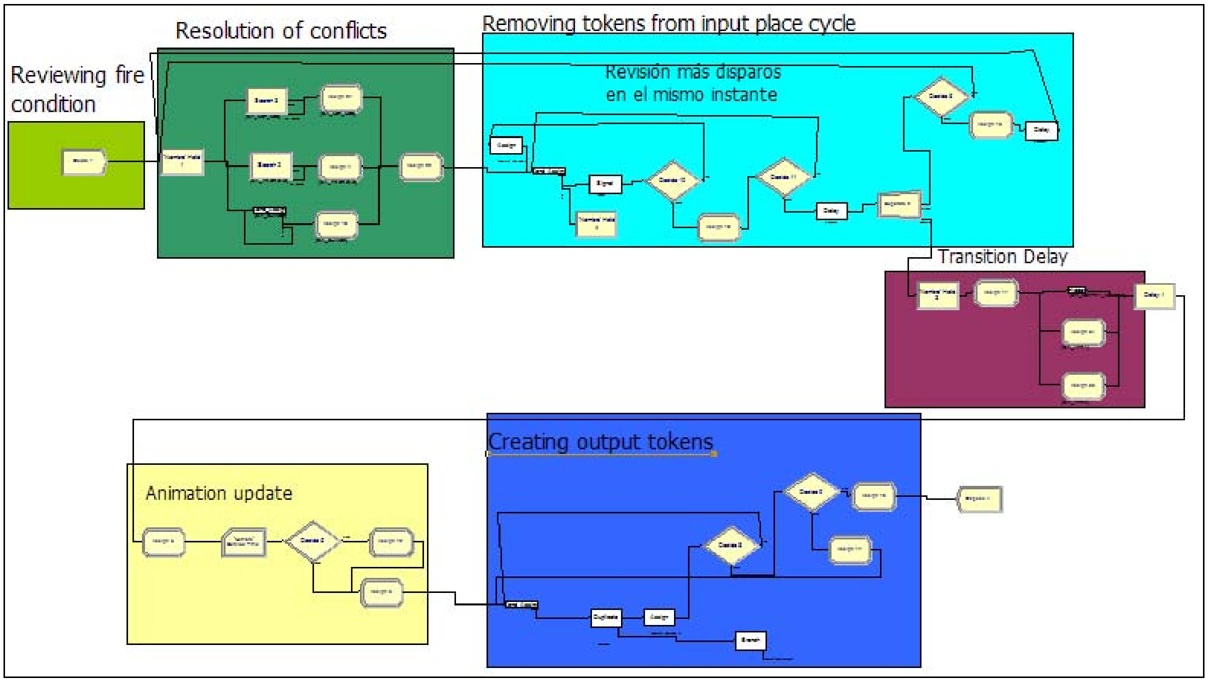

Figure 4: Colored transition modeling

Another construct is the Conflict module. The purpose of this module is to select a transition to fire among those which are enabled at a given time. Its logic has the following processes.

1. Firing Evaluation: Contains the conditions set for firing a transition
2. Condition Max/Min: Similar to the queue sort rules of ARENA. The transition to fire among those in conflict is defined according to a firing rule. Such firing rule is set according to a Max/Min criterion. For example the transition with the most input places (Max) or with the minimum firing time (Min).
3. Chance: If the user selects this option, transition conflicts are resolved with firing probabilities. Such probabilities define among the enabled transitions which one fires. The firing probabilities are user-defined.

4 CASE STUDY: CREATIVE COLORS

Creative Colors is a company that manufactures and finishes make-up pencils. The manufacturing processes are engraving, screen, sharpening, assembly, thermo-shrink wrapping and packing. The product mix is very diverse as the company manufactures about 100 different product references. In addition, eight process routes were identified for an equal number of families of products. The sales orders come from different customers in varying quantities.

One of the problems that Creative Colors is currently facing is reducing the cycle times. The service level (fill rate or average percentage of orders met from stock) was about 50%. This caused major concerns among the major shareholders who face stiff competition from companies in Panama and Costa Rica. In addition, the company had high overtime costs and loss of goodwill.

In this study, a CONWIP (CONstant Work In Process) policy was designed (Hopp and Spearman 1997). This policy aims to keep constant amounts of semi-finished products (WIP) at the production lines. The CONWIP policy only authorizes new items to the system if the predefined value of the WIP has not been surpassed.

The purpose of the simulation was to study the effect adding new technicians and the impact of adding new machines. Creative Colors currently has the following machine park as shown in Table 2:

Table 2: Available Resources at Creative Colors

Machines	Number of Resources
Engraving Press	4
Screen Machine	2
Sharpening Machine	1
Tapping Machine	4
Thermo-shrink wrappers	6

The engraving presses, screen and the thermo-shrink wrapping machines undergo long changeover operations and the corresponding changeover times are dependent on the processing sequence. The other processes do not require significant setups.

These setup operations are currently performed by three technicians with no significant performance differences between them. Additionally, on each machine there is a person that checks and classifies the number and quality of the outgoing pencils.

The machines work at fix speeds that are dependent on the specific items. In the case of engraving, settings that can affect the processing time are tape quality, die quality and temperature.

During the execution of the manufacturing processes, some machines are usually recalibrated and therefore the effective processing times are higher than those calculated with the machine output rates. Based on three-month observations, the probability distributions for both the setup and nominal unit processing times are shown in Table 2. The productivity ratios (nominal unit time / effective unit time) are also given in Table 3.

Table 3: Processing Data

Process	Productivity	Setup time (min)	Unit Processing Time (sec / unit)
Engraving	76%	UNIF (21,43)	TRIA (2, 2.5, 3)
Screen	94%	UNIF(7,25)	NORM (1.55, 0.2)
Sharpening	98%	TRIA(0.75, 1,1.25)	UNIF (0.5, 0.7)
Assembly	97%	UNIF(25,30)	1.5 + 2BETA(1.4, 1.6)
Thermo-shrink wrap	74%	TRIA(3,17, 39)	1.5 + GAMM(0.1, 2.9)

As it was mentioned above, products were classified in eight different families and each family followed a manufacturing route. All product families are not sold in equal quantities and therefore a probability of occurrence (sale) was assigned to each family. The routes and probabilities (according to the sales mix) for each product are shown in Table 4.

Table 4: Routes and Probabilities

Routes	Probabilities
Engraving, sharpening, assembly	36%
Engraving, sharpening, assembly, thermo-shrink wrap	21%
Screen, sharpening, assembly	13%
Screen, sharpening, assembly, thermo-shrink wrap	13%
Engraving, sharpening, assembly, thermo-shrink wrap	7%
Screen, thermo-shrink wrap	6%
Screen, screen, thermo-shrink wrap	4%

There are additional factors in the process that usually involve workers. Such processes include part counting and quality tests (go-no go), report elaboration and preparation of raw materials.

The following assumptions were made:

- Raw materials, resources are 100% available during the production runs.
- Machines may suffer loss of productivity and such loss implies delays in the delivery times.
- Each product follows one route during its execution at the shop floor.
- Jobs cannot be interrupted (no pre-emption).
- Resources can only process a job at a time.

4.1 Petri Net Simulation

A Colored Petri Net model was developed for the above manufacturing plant using the HPNM template described above. Figure 5 shows the proposed model of the system.

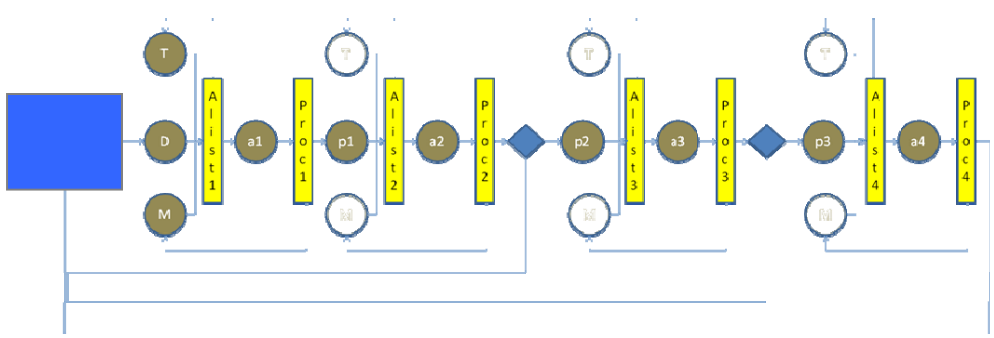

Figure 5: Creative Fab Model

A number of simulation tests were run. The idea was to use different WIP levels and observe the impact on the cycle time and throughput. Recall from the basic Little Law that there exists a "critical" WIP level that is the minimum WIP level in which the maximum production rate is attained (Hopp and Spearman 1997). In very simple systems this can be easily evaluated; simulation is required for real life complex systems. Based upon the results shown in Figure 6, a realistic maximum daily production rate was taken as 45000 units per day. For this production rate, the minimum WIP level was defined as 80000 units.

Figure 6: Throughput rate vs WIP

Figure 7 illustrates the average cycle times vs. the WIP level. The curve is relatively flat for the WIP levels up to 50000 units and then increases at a rate of 1.5 hours per 10000 units. This observation conforms the classical plant dynamics expressed by the Little's Law.

Once the WIP level is determined along with the expected throughput and cycle time, the next step was to study the effect of adding technicians and machines.

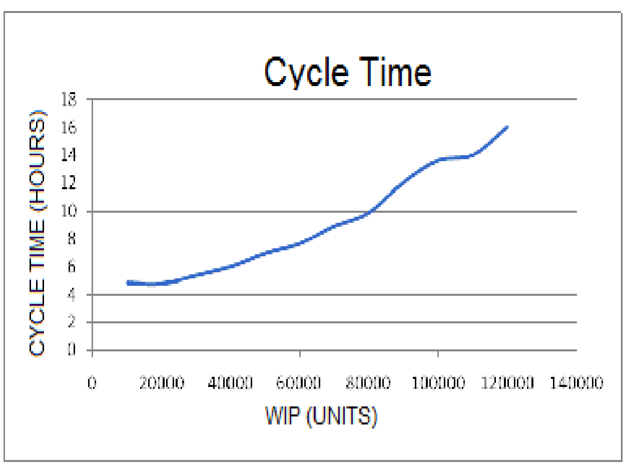

Figure 7: Average Cycle Time

4.1.1 Technicians

The number of technicians was varied from two to five. The simulation shows that increasing the number of technicians (from three to four and five) does increase neither the cycle time nor the throughput. See Figure 8. On the other hand, eliminating a technician does affect the system performance. Although the cycle time is reduced to about 9.5 hours, the throughput rate decreases in 5500 units per day.

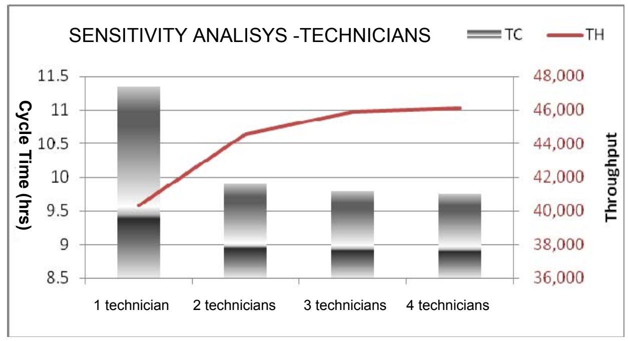

Figure 8: Sensitivity Analysis for Number of Technicians

4.1.2 Machines

The addition of new machines was also evaluated. The first step was to determine the bottleneck machine which in this case was the engraving machine. With this bottleneck machine in mind, the number of machines was progressively increased. The following scenarios were tested and validated with the company's management:

- One extra engraving machine (1E)
- Two extra engraving machines (2E)
- Three extra engraving machines (3E)
- Two extra engraving + one extra sharpening machines (2E + 1T)
- Three extra engraving + one extra sharpening machines (3E + 1T)
- Two extra engraving + one extra sharpening + 1 extra screen machines (2E + 1T + 1S)
- Two extra engraving + one extra sharpening + 1 extra screen + 1 assembly machines (2E + 1T + 1S +1 cap)

Figure 9 shows the daily maximum production rates. The higher increase was achieved by adding one engraving and one sharpening machines (10000 additional units or a 20% increase over the current rate). The maximum rate was achieved with the 2E+1T+1S+1Cap but it does not seem to be a feasible alternative as the investments would be very high.

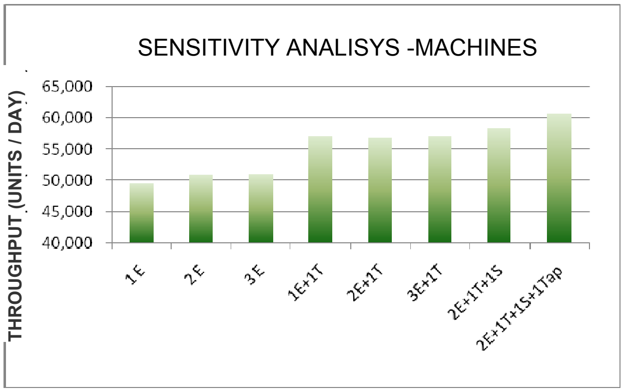

Figure 9: Sensitivity analysis for machines

5 COMPARISON OF METHOLOLOGIES: DES VS. TCPNS

A major issue is the comparison of the modeling with classical DES and with Petri Nets. The above plant was also modeled via DES on the ARENA® software about 2 years before the TCPN model. This paper analyzes issues from qualitative and quantitative perspectives.

a) Running Time

The results in terms of performance indicators were very similar, keeping in mind that the DES model included a few details that are very difficult to incorporate within the Petri Net model. The running time for the DES was on average 16 minutes whereas the average running time for the TCPN was 2 minutes.

b) Lines of Code (SIMAN)

Each line of code in SIMAN represents an instruction. The number of lines of code with the DES model was about 6900 whereas the TCPN involved only about 1000 lines of code. This fact shows that the TCPN modules were highly effective in keeping the code compact.

c) Construction of Scenarios

In the experience of the authors, constructing scenarios with DES is complex since it uses simple modules that are combined and changing the logic of such modules is time consuming and error prone. On the other hand, the use of high level coding as in the case of the TCPN template facilitates the creation of new scenarios. Changing the scenarios in the DES model took about a week whereas with HPNM the same process took about one day and a half.

d) Modeling power

The modeling power of DES outperforms that of the TCPNs. Many situations such as continuous processes, job, pre-emptions, time related decisions, and access to global information are hard to model with TCPNs and are easily created with DES.

e) Model Development

In terms of the model development, the TCPN project was finished in about four days as compared with the one month period required for the DES model. This development time did not include the data acquisition, team meetings or activities not related to coding and running. Although it is clear that there was a learning period for the first DES development, the HPNM seems to be faster to adapt. The reason may be the standard and widely studied modeling techniques for TCPNs.

6 CONCLUSIONS

This paper has presented HPNM which is a library griten with the aid of the ARENA® templates. The library uses the common ARENA® blocks to create the basic constructs of Timed Colored Petri Nets.

The proposed approach combines the modeling power of DES with the Petri Net formalism. The combination of DES and Petri Nets meant savings in data structures, parameter inputs, lines of code and modeling and running efforts.

The proposed approach seems very appropriate for systems in which the level of detail is not "very high". The system modeled with HPNM seems appropriate for this tool.

Petri Nets have been mostly used for short-term decisions, usually at the operation level (control of operations). In this paper, Petri Nets were used for higher level tactical-strategic decision such as decisions on the number of machines and workers. With the help of DES, Petri Nets can be leveraged at higher decision levels.

The case study presented in this paper showed the feasibility of using TCPNs in real systems at the tactical level and this fact has not been very common in the literature. In fact, their mature modeling methodologies and ease of use may speed up the development of industrial simulations without losing the capability analysis of DES software.

REFERENCES

Dubois, D., and K. E. Stecke. 1983. Using Petri nets to represent production processes. In *Proceedings of the 22nd IEEE Conf. on Decision and Control,* 1062 – 1067. San Antonio, TX.

Desrochers, A., and R.Y. Al'Jaar. 1995. Applications of Petri Nets in Manufacturing Systems: Modeling, Control and Performance Analysis. 1st edition, IEEE Press..

Hopp, W., and M. Spearman. 2000. *Factory Physics*. 2nd ed. New York: McGraw-Hill, Inc.

Jensen, K. 1981. Coloured Petri Nets and the Invariant Method. *Theoretical Computer Science* 14: 317-336

Kelton, W. D., R.P. Sawdowski, and D. T. Sturrock. 2004. *Simulation with Arena*. 3rd ed. New York: McGraw-Hill, Inc.

Law, A. M., and W. D. Kelton. 2000. *Simulation modeling & analysis*. 3rd ed. New York: McGraw-Hill, Inc.

Mejía, G., and N. Odrey. 2005. An Approach Using Petri Nets and Improved Heuristic Search for Manufacturing System Scheduling. *Journal of Manufacturing Systems* 34 (2):79-92.

Zhou, M. C., and F. DiCesare. 1993. *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems.* Kluwer Academic Publishers.

AUTHOR BIOGRAPHIES

GONZALO MEJÍA is an associate Professor at the Industrial Engineering Department of the Universidad de los Andes in Bogotá, Colombia. He obtained his PhD from the Industrial Engineering Department at Lehigh University in 2002. His research interests include the modeling and optimization of production systems with Petri Nets. His papers have been published in important journals such as the Annals of Operations Research and the International Journal of Production Research.

DIEGO MARTÍNEZ graduated from the masters program at the Industrial Engineering Department of the Universidad de los Andes in Bogotá, Colombia. He is a senior consultant in the private sector for manufacturing and services companies. He has been responsible a number of simulation projects for Latin American Companies.

FIDEL TORRES is an associate Professor at the Industrial Engineering Department of the Universidad de los Andes. His research interests include logistics and production. He is the director of the PyLO (Production and Logistics Group) at the Universidad de los Andes.