Davidrajuh, R. (2012).) Activity-Oriented Petri Net for scheduling of resources. 2012 IEEE International Conference on Systems, Man, and Cybernetics. October 14-17, 2012, COEX, Seoul, Korea

Activity-Oriented Petri Net for scheduling of resources

Reggie Davidrajuh

Abstract:
This work presents a new methodology for modeling discrete systems in which activities compete for resources. The new methodology known as Activity-Oriented Petri Net (AOPN) allows modelers to concentrate on modeling activities, and the resources are assumed to be managed by the underlying system; this approach allows compact and much simpler models even for large and complex systems. General Purpose Petri Net Simulator (GPenSIM) is a realization of AOPN methodology on MATLAB platform. In addition to comparison of AOPN methodology with the other existing methodologies, a case study is also provided in this paper to better explain how an AOPN model of a resource scheduling problem can be modeled and simulated with GPenSIM.

Introduction

Scheduling of system resources is an active research topic for many decades now. Petri net is widely used for modeling and simulation of scheduling problems as Petri net effectively model concurrency and conflict [16].

The core issue of the scheduling problem is the optimal allocation of resources so as to perform all the required activities. The optimal allocation of resources generally refers to minimization of costs and the time to complete the activities. Typical discrete event dynamic systems (DEDS) where resource scheduling takes place often are computer multitasking environment, flexible manufacturing systems, and logistics.

In this paper, a new methodology known as Activity-Oriented Petri Net (AOPN) is presented. A new tool, known as General purpose Petri Net Simulator (GPenSIM) is designed to realize AOPN models; GPenSIM is also described briefly in this work. AOPN emphasizes “activity-based modeling”. The few literature that focus on “activity-based modeling” are within transportation and logistics area; e.g. [10][11]. There are also a few studies that uses activity-based modeling in production systems and manufacturing area: e.g. [8][9]. This paper on AOPN is a continuation of [3][4], takes activity-based modeling approach for modeling of resource scheduling problems, uses Petri nets, and simplifies the models by pushing resource management into background.

Activity -Oriented Petri Net

There are two types of resources:

- Primary resources: resources for part production and value added services (like production machines), and
- Secondary resources: resources for part handling and movement of material (like conveyor belts, robots, lifts, AGVs).

In the following subsections, two existing methodologies for modeling scheduling problems with Petri nets are discussed; these methodologies are Process-oriented Petri net (POPN) and Resource-oriented Petri net (ROPN). Then, these two methodologies are compared with the new methodology AOPN that is proposed in this work. In order to explain the differences between these three methodologies the following problem will be modeled by the three methodologies.

Scheduling Problem in a flexible manufacturing system

A flexible manufacturing system (FMS) consists of two CNC machines, one for vertical machining (VNC) and one for horizontal (HNC) machining. FMS also has a robot for movement of parts and material between these CNC machines, and between the machines and input and output buffers.

FMS can perform three types of jobs:

- Job-A uses only the VNC machine
- Job-B uses the VNC first and then the HNC machine, while holding the robot throughout
- Job-C uses only the HNC machine

A. Process-Oriented Petri Net (POPN)

Process-oriented Petri nets (POPNs) is a widely used methodology for modeling discrete systems that involves resources [13]; POPN models are obtained by following the processes that make up the system. POPN models are simple to understand and the connections between the elements (places and transitions) of the model resemble the topology of connection that exists between the elements (e.g. machines and resources) in the physical system. However, one of the disadvantages of the POPN models is that they can be large, sometimes too large as all the resources and all their connection with the other elements in the system, is shown in the model, causing extra large number of places, transitions, and arcs [6][14][15]. It will be difficult to analyze and control POPN models, because of the complexity due to the large size of the models [13]. Another disadvantage of POPNs is that they are prone to deadlocks. In a POPN, arcs that connect the resources with the other elements induce cycles (called “resource circuits” in [13]); [13] proves that the cycles introduce ill-behaved siphons (places) that are emptied of tokens during simulations causing deadlocks.

Figure-l shows the POPN model of the scheduling problem in FMS. The POPN model has three modules (module-A, B, and C) for the three jobs, connected together with the places that indicate availability of the machines VNC, HNC, and the robot. The POPN model is straightforward showing all the elements (machines, robot, input and out buffers) and the connections between them.

It is visible from figure-l that the places represent both the availability of the resources and also the status of the activities. E.g. place ‘pR’ and ‘pHNC’ represents the availability of the resources robot and HNC machine respectively; the place ‘pA.in_VNC’ indicate the movement of material from the input buffer towards the VNC machine, and ‘pA.VNC’ indicate that the machining is undergoing at the VNC machine. Transitions play only a utility role, representing switching between the activities.

B. Resource-Oriented Petri Net (POPN)

ROPN is proposed for modeling manufacturing systems in which different manufacturing activities compete for resources [12]. In ROPN, each resource is modeled by a single Petri net place; the different manufacturing processes that compete for resource are modeled separately as subnets, and then the subnets are forged together to make the overall model. When forging, the subnets are fused together so that the places that represent a specific resource in different subnets become a single unique place in the overall model [12]. Hence, in the overall model, the different resources are modeled by a place each, makes the model compact.

Since a place in POPN represent a specific resource that can be used by different activities at different times (e.g. for making different kinds of parts), coloring is used to differentiate the part flow (token flow); thus ROPN methodology necessitate the use of Colored Petri net, whereas POPN does not.

In ROPN, transitions play more role than in POPN: Figure-2 shows the ROPN model of the scheduling problem. In figure-2, it is very clear that the places represent only the availability of the resources; transitions play more role in ROPN than in POPN, as transitions represent the activities; e.g. transitions ‘tARl’ and tRA2’ represent the activity of moving material from the input buffer towards the VNC machine, and the machining at the VNC machine respectively. This means, in the POPN model transitions ‘tAPl’ and ‘tAP2’ which represent the start and the completion of the movement of material from input buffer into VNC machine, and the place ‘pA.in_VNC’ which represents the status on on-ging activity, are forged into a single transition ‘tARl’ in the ROPN model.

Though by the use more active transitions, reduction in complexity and in the size of the ROPN model is achieved, ROPN methodology does not answer some other situations:

- What if an activity needs *some* specific resources (let say, an activity needs two machines *mi* and *mj*, in addition to a robot) *at the same time?* What if an activity demands one specific machine *mi* and *any other machine* that is available, at the same time?

ROPN methodology appeases only modeling of activities that use *at most one resource at a time*.

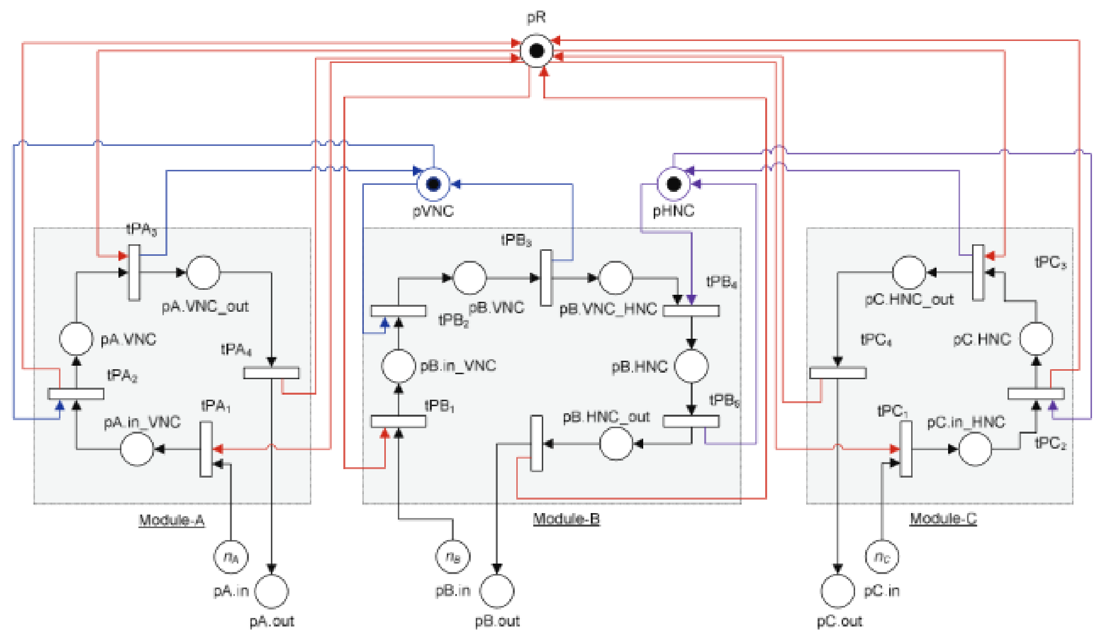

Fig.1. POPN model of the FMS

C. Activity-Oriented Petri Net (AOPN)

Activity-Oriented Petri net (AOPN) can be considered as a natural extension to ROPN; AOPN is devised to support an activity demanding *any number of resources*.

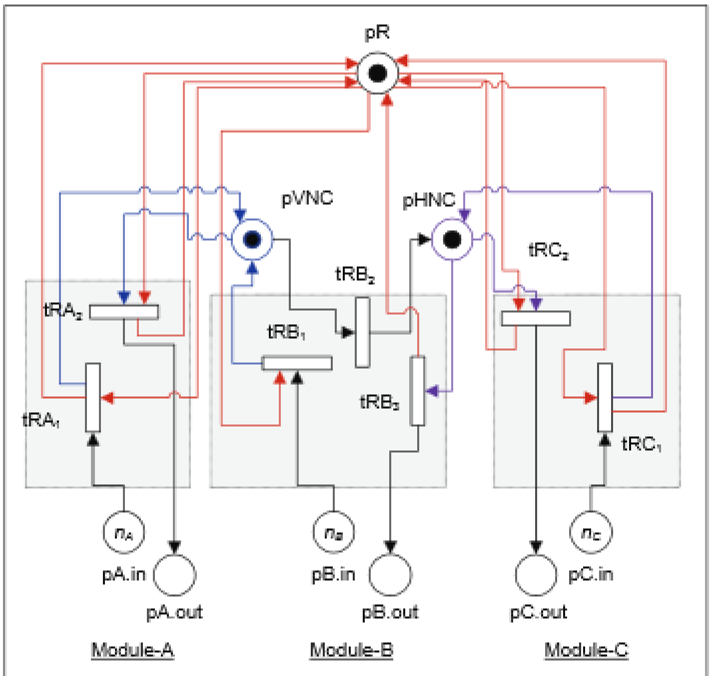

Fig.2. ROPN model of the FMS

AOPN methodology abstracts way both primary and secondary resources; AOPN methodology only considers the activities during model building, and the resources that are used by the activities are pushed into the background. AOPN is based on the following rationale:

- Both POPN and ROPN methodologies are "resource-based modeling" methodologies emphasizing resources and their usage. Resources take the central stage in the models, making the resources very visible and the activities are seen sewn around the resources.
- The resource-based modeling methodologies are the default approaches for scheduling of DEDS; this is because, in a real-life DEDS environment, it is the resources (robots, machines, humans, computers, etc.) that are visible and tangible to humans, whereas activities are generally intangible and sometimes invisible e.g. electronic activities. Thus, we humans are tended to emphasize resources, intentionally or unintentionally, in our modeling methodologies.
- We humans are used to think "geometrically", focusing mainly on the elements we see and feel, taking into consideration of how these elements are connected together. Thus, resource-based modeling tends to become the natural way of developing model of DEDS.
- Resources are also the expensive elements in an engineering environment, thus investors try to get their money back (return on investment) faster; thus, the goals of the modeling approaches become optimal use of the resources rather than optimal execution of the activities.

In AOPN methodology, emphasis is given to activities on the ground that it is the activities that produce something or add value; resources are passive and are worthless unless they are utilized by some activities:

- The activities may reserve, use, and then release resources during their operations; resource management is done transparently by the underlying system in the background
- There is no differentiation of resources: both primary resources and the secondary resources are considered as system resources

- Group of activities that belongs to a job are modeled separately as a subnet, independently from other activities that belongs to the other jobs. Then, the subnets are put together as the overall model, apparently the subnets are unconnected and in parallel.

Figure-3 shows the AOPN model of the FMS example. In this model, only the activities are modeled as transitions; resources are implicitly modeled within the transitions. For example, transition ‘tAAl’, will *try to acquire* (reserve) the robot for movement of material from the input buffer into the VNC machine. Only if the transition ‘tAA1’ is granted the robot then it will start operating (transition ‘fires’). After completion, the transition ‘tAAl’ will release the robot for other activities. Thus, in ROPN model, the transition ‘tAR1’ is forged with the place ‘pR’ to become the transition ‘tAA1’ in AOPN. Since in AOPN, the resources are contained within the transitions that represent activities, there is no need for explicit representation of resources by places; this eliminates not only places representing resources from the model and also all the connections between the resource places and the relevant transitions.

D. Summary: Comparing POPN, ROPN, and AOPN

For a resource scheduling problem consisting of a total of *n* number of activities (belonging to various jobs) and *r* number of resources, the table below shows the size of Petri net models obtained by the three methodologies.

	POPN [13]	ROPN [13]	AOPN
Number of places	$O(n+r)$	$O(r)$	$O(n)$
Number of transitions	$O(n)$	$O(r^2)$	$O(n)$

In addition to the linear size of the AOPN models, the most important advantages of AOPN over POPN and ROPN is that modeling of the activities that demand composite resources (more than one resource) is possible in AOPN: this advantage is due taking away resource management complexities from the Petri net model and to pushing into the underlying resource management system (RMS), which is explained in the next sections.

GPENSIM AS a REALIZATION OF AOPN

The complexity of many discrete systems (e.g. in computer operating systems, manufacturing systems, logistics) is mainly due to the resource management; resource management basically deals introduction and deletion of resources into the resource pool, and then managing reservation, allocation, and redemption of resources by clients.

A. System Resource Management

The previous section shows that the AOPN methodology expects the underlying system to manage resources in the background, just as computer operating systems provide main memory management for completing parallel programs. AOPN methodology expects the underlying system to do the resource management that can be classified as follows:

- Management of common resources: a client need not name a resource while requesting for a resource. E.g.: a CPU can acquire *any* communication channel (out of three communication channels that are available) to send a file to the printer

- Management of named resources: a client must name a resource while reserving. E.g. in *a* car mechanic shop, mechanics ‘A1’ and ‘Chuck’ are needed to fix a car.
- Management of multiple instances of a specific resource: a resource can have many instances, and a client can request one, many, or all of the instances. E.g. *a* computer file can be opened to be read by a limited number of computer programs; however, the file must be closed and allowed to be used by only one program if the program writes on the file.

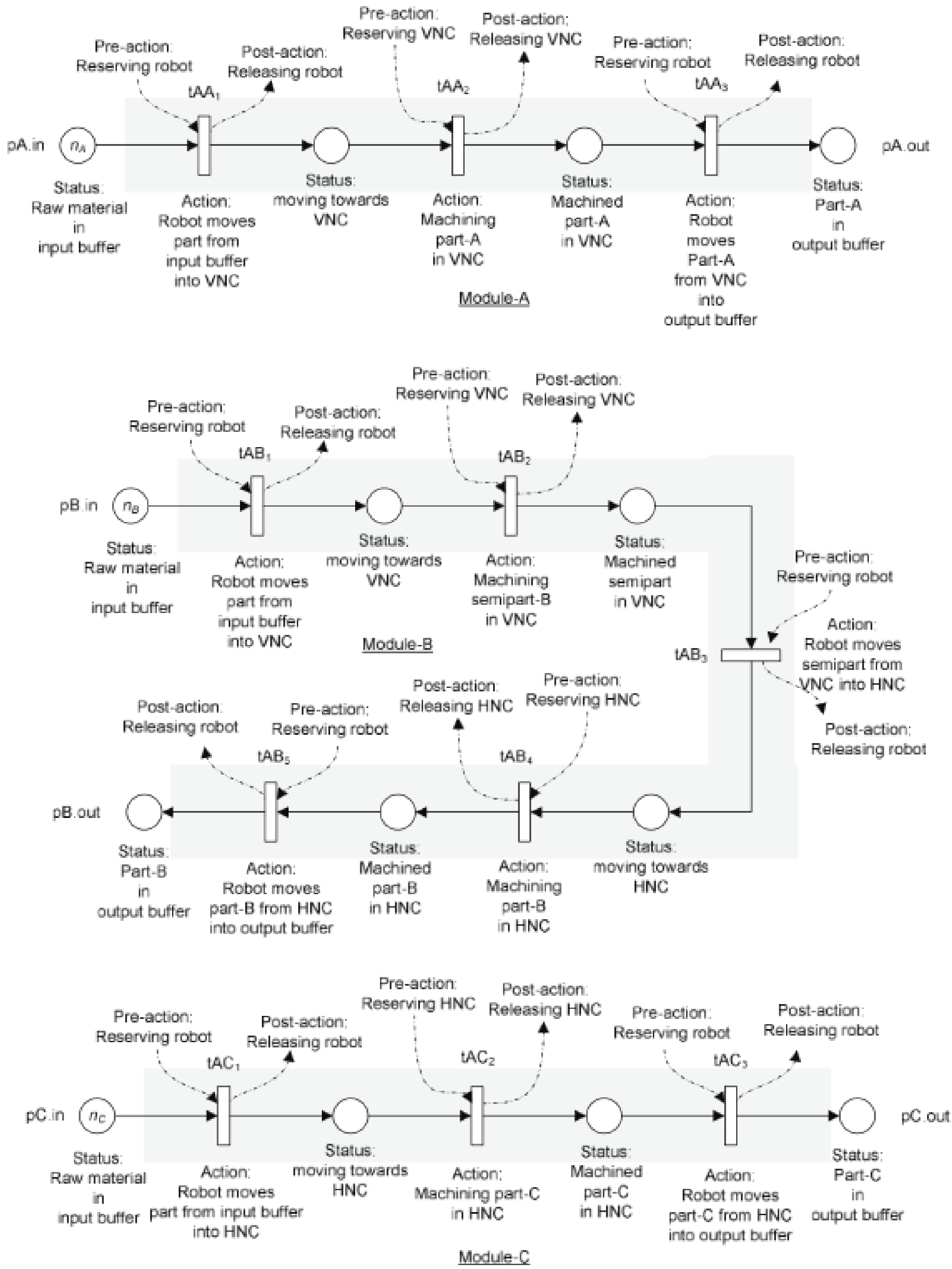

Fig.3. AOPN model of the FMS

General propose Petri Net Simulator (GPenSIM) is a realization of AOPN methodology that is available for MATLAB platform [5]. GPenSIM provides the resource management so that a modeler following AOPN methodology needs only to concentrate of how the jobs (consisting of many activities) can be modeled individually, one at a time, and then combined together to make the complete model.

B. GPenSIM

GPenSIM is a tool for modeling and simulation of discrete-event dynamic systems (DEDS) [1][2][5]. GPenSIM is based on Petri nets. The reasons for developing a new simulator are:

For basic users: A tool that is easy to understand and easy to use, even for users with limited mathematical and programming skills

For advanced users:

- allow seamless integration of DEDS models with the other toolboxes that are readily available on the MATLAB platform;
- allow easy extension of GPenSIM functions
- Compact simulation code

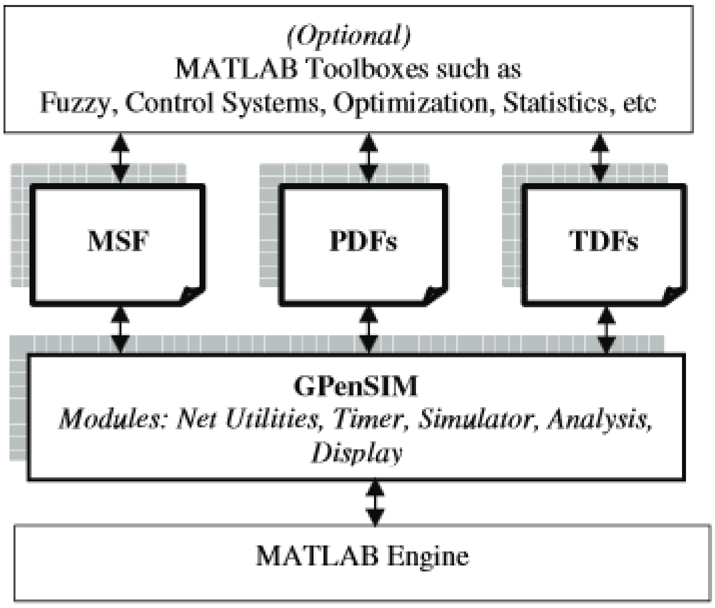

Fig. 4. The files for simulation

GPenSIM is developed by the author of this paper, in order to satisfy the four criteria stated above (ease of use, flexible, extensible, and compact code). GPenSIM is realized as a toolbox for the MATLAB platform, so that diverse toolboxes that available in the MATLAB environment (e.g. Fuzzy Logic Toolbox, Control Systems Toolbox, Advanced Statistical Analysis Toolbox, etc.) can be used in the discrete models that are developed with GPenSIM.

A DEDS model created GPenSIM has 3 components (figure-4):

- Petri net Definition File (PDF): the elements such as places and transitions of a Petri net model, and the static connections that exist between the elements (arcs) are defined in PDF
- Main simulation file (MSF): initial dynamics (initial tokens, etc.) to the system are defined in MSF, and
- Transition Definition File (TDF): the interaction between the Petri net model and the environment are defined in TDF file(s)

Interested reader is referred to [5].

Case Study

A FMS consists of a robot ('R') and a machining center ('MC') with 3 cells; the cells are identical and can perform operations in parallel. After a total of 20 operations, the machining center must undergo a maintenance cycle.

There are three types of jobs, job-A, job-B, and job-C:

- Job-A consists of two activities: activity-A1 needs two instances of MC (that is, 2 cells are needed). After completion of activity A1, activity A2 takes over the two cells and continues to use it. At the completion of A2, the cells are released.
- Job-B consists of three activities: to start with, activity-B1 needs the robot and one instance of MC. After completion of activity B1, activity B2 takes over the cell from B1 land continues to use it. In addition, B2 needs one more cell to start. Right after the completion of B2, activity B3 starts and continues to use the cells obtained from B2. These cells will be released at the completion of B3.

Jobs A and B each needs two cells. However, there are only three cells, thus these two jobs cannot be executed in parallel.

- Job-C is the maintenance cycle that consists of only one activity C1 - "the maintenance activity" of all the cells at the machining center MC. Activity C1 will start when the number of times cells are used becomes more than 20. Since jobs A and B are always competing for the cells, the maintenance activity has high priority, so that it gets the cells for maintenance whenever the maintenance cycle has to be started.

Figure-5 shows the complete AOPN model of the FMS. It will be not possible to model the problem with *POPN/ROPN* methodology as:

- POPN does not support assigning higher priority to 'tC1'; unless 'tC1' gets higher priority, it cannot grab all the cells and perform maintenance when the cells become overused.
- ROPN methodology does not support activities acquiring more than one resource at the same time.

It is apparent from figure-5 that the APON model is very simple and compact, and it clearly represents the activities of the three jobs A, B, and C. The code for the simulation is also simple and compact: the complete code for simulation takes only 120 lines of GPenSIM code, divided into 9 simple files. Due to brevity, the code is not shown here, but the interested reader is encouraged to inspect the code at [7].

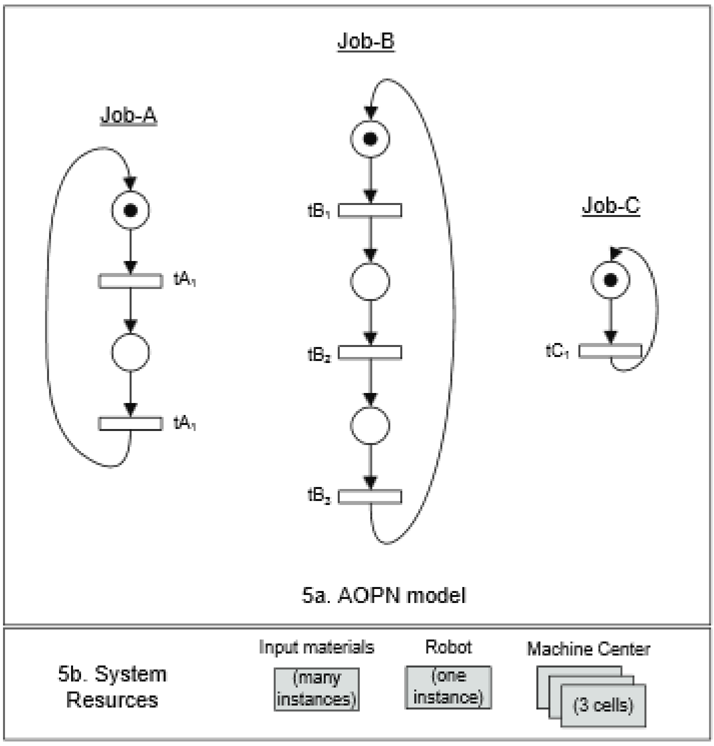

Fig.5. AOPN model and the system resources for the case study

The simulation results are given below:

Resource Usage Summary:

Number of resources: 2
Simulation start time: 0
Simulation completion time: 48

Line Time = 244
Total time at resources: 162
Resource Usage (Line Eff.) = 66.3934 %

Resource *** R ***: Total Time spent at R: 40
Used by activity "tB1": Time spent: 40

Resource *** MC ***: Total Time spent at MC: 122
Used by activity "tA1": Time spent: 42
Used by activity "tB1": Time spent: 40
Used by activity "tB2": Time spent: 20
Used by activity "tC1": Time spent: 20
**

Conclusion

This work presents a Petri net based modeling methodology known as Activity-Oriented Petri Net (AOPN) for modeling discrete systems that involves with common resources; AOPN modeling methodology emphasizes activities with the assumption that it is the activities that are active in generating value and that the resources are passive sitting idly to be exploited. AOPN maps the activities into transitions of a Petri net model with intermediate places as buffers between the transitions. This demands an external system to manage distribution of resources between the competing transitions; GPenSIM provides this support.

References

Davidrajuh, R. and Lin, Binshan (2011) "Exploring Airport Traffic Capability Using Petri Net based Model", Expert Systems With Applications (ESWA), 38 (2011) pp. 10923-10931

Davidrajuh, R. (2011) "Extending a Petri Net Simulator as a Real-Time Control Simulator'", IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2011), Penang, Malaysia, November 25-27, 2011

Davidrajuh, R. (2011) "Representing Resources in Petri Net Models: Hardwiring or Soft-coding?", 2011 IEEE International Conference on Service Operations and Logistics, and Informatics, Beijing, China, July 10-12, 2011

Davidrajuh, R. (2011) "Scheduling using 'Activity-based Modeling'", 2011 IEEE Conference on Computer Applications & Industrial Electronics (ICCAIE 2011), Penang, Malaysia, December 4-7, 2011

GPenSIM user Manual (2012) Available at: http://davidrajuh.net/gpensim

Jeng, M. D., and F. DiCesare. 1995. Synthesis using resource control nets for modeling sharedresource systems. IEEE Transactions on Robotics and Automation 11:317-27

Resource Scheduling Example (2012): Available: http://www.davidrajuh.net/ gpensim/resource-scheduling-SMC2012

Spedding, T. and Sun, G. (1999). Application of discrete event simulation to the activity based costing of manufacturing systems. Int. J. Production Economics 58 (1999) 289-301

Tatsiopoulos, I and Panayiotou, N. (2000) The integration of activity based costing and enterprise modeling for reengineering purposes. Int. J. Production Economics 66 (2000) 33-44

Wang, D. and Cheng, T. (2001) A spatio-temporal data model for activity-based transport demand modelling. Int. J. geographical information science, 2001, vol. 15, no. 6, 561-585 (Pubitemid 32818725)

Witlox, F. and Tindemans, H. (2004) The application of rough sets analysis in activity-based modelling: Opportunities and constraints. Expert Systems with Applications 27 (2004) 585-592

Wu, N. Q., and M. C. Zhou. 2007. Shortest routing of bi-directional automated guided vehicles avoiding deadlock and blocking. IEEE/ASME Transactions on Mechatronics 12:63-72. (Pubitemid 46444219)

Wu, N. and Zhou, M. (2010) System Modeling and Control with Resource-Oriented Petri Nets. NY: CRC Press

Zhou, M. C., and F. DiCesare. 1991. Parallel and sequential mutual exclusions for Petri net modeling of manufacturing systems with shared resources. IEEE Transactions on Robotics and Automation 7:515-27 (Pubitemid 21674694)

Zhou, M., F. DiCesare, and A. Desrochers. 1992. A hybrid methodology for synthesis of Petri nets for manufacturing systems. IEEE Transactions on Robotics and Automation 18:350-61.

Zhou, M. C., and K. Venkatesh. 1998. Modeling, simulation and control of flexible manufacturing systems: Petri net approach. Singapore: World Scientific.