

An Approach for Project Scheduling Using PERT/CPM and Petri Nets (PNs) Tools

Amer A. M. Boushaala

(Department of Industrial and Manufacturing Systems Engineering / University of Benghazi, Libya)

ABSTRACT: The Petri Nets are more powerful models as compared to the PERT/CPM charts. The Petri Nets show the same type of scheduling constraints as the PERT/CPM charts does. We can easily convert a PERT/CPM networks to a Petri Net model. This work shows how to use Petri net's for modeling and verifying project management networks. Software for a functional analysis of a Petri Net model is developed. The developed software enables analyzing of a system for its various properties irrespective of time. The software analysis is based on some approaches used to determine the various properties of PN model: liveness, safeness, reach-ability and conservativeness. This paper proposes a PN based modeling approach to provide a formal way to verify that all the activities are well connected in the project network. This PN model can then be analyzed through (PN based) scheduling techniques, used to find the critical path, used as a basis to develop algorithms for resource-constrained project management, and used for other decisions related to project management. The proposed PN based approach enables the project manager to consider not only resources but also all different types of variables/constraints related to a project, e.g. costs of activities.

Keywords: CPM, PERT, Petri Net's, Verifications

I. INTRODUCTION

CPM/PERT are major tools for project management. They are networks that show precedence relations of activities of a project with the activity times. They are used to find project completion time, which is the longest path in the network. They are also used to find other related information about the activities, e.g., slack time, earliest start and end time. All these analyses are made based on the infinite resource assumption. Then, resource allocation of the project is carried out. That is, finite resources are allocated to the project activities whose time requirements were calculated based on the infinite resource assumption. However, instead of this two-step approach to the resource-constrained project management, a one-step approach, through Petri nets (PNs), is possible. Furthermore, CPM/PERT; do not provide a formal way to verify that all the activities are well connected in the project network.

Liang, Chen and Wang [1] introduced a project model, called SPREM, which extends CPM/PERT's notation to four types of vertices to express the non-deterministic and iterative behaviors of software engineering projects. They compared it with PNs, and discussed its analysis and behavioral properties. These properties are verified through several algorithms they proposed. However, as discussed in section 5, the well known place (transition) invariants help verify some of these properties without any additional algorithms.

Desrochers and Al-Jaar [2] give some advantages of PNs: PNs capture the precedence relations; conflicts and buffer sizes can be modeled easily; PN models represent a hierarchical modeling tool with a well-developed mathematical and practical foundation. These advantages help model and verify (resource-constrained) project networks.

Several PN based approaches to project management were proposed in the literature. Jeetendra et al. [3] evaluate conventional project management tools, state their inadequacies and list advantages of using PNs for project management. Chang and Christensen [4] propose PM-Net for software development, which adopts the basic concepts of PNs with extensions to represent both decisions and artifacts. Ashok Kumar and Ganesh [5] also describe project management models with their inadequacies, and propose a new model based on PNs and demonstrate its usefulness for real-time activity scheduling in a resource-constrained project environment. However, these approaches need to modify and extend basic PN semantics and add new tools to analyze the project network. For example, Jeetendra et al. [3] change PN semantics through a precedence matrix they developed. Conflicting transitions in traditional PNs behave concurrently in their definition. This matrix is then used to help determine floats and the critical path and to find deadlocks. Ashok Kumar and Ganesh [5] also add several features to basic PN semantics for project management.

The proposed PN based approach facilitates modeling (resource-constrained) projects, and verifying some properties of the projects networks, exploiting the well known place (transition) invariants, through the basic PN definitions. That is, contrary to some approaches in the literature, there is no need to modify the basic PN semantics to model activities and decisions related to a project. This increases the capability of the basic PNs, without any extension in the theory, to model projects. Transitions are used to model all the preconditions, for an activity to start, which is modeled via a place. The place invariants are then used to verify some properties of the projects, as discussed in section 5. This model can then be analyzed through (PN based) scheduling techniques [6, 7], and used to find the critical path [8], and used as a basis to develop algorithms for resource-constrained project management [9, 10]. These considerations are beyond the scope of this paper. The aim of this paper is to show how to model project networks with PNs without any modification or extension in the basic PN semantics, and how to compute the critical path and verify some project properties through the well known place invariants.

II. PERT/CPM ACTIVITY NETWORKS

The foundation of the approach came from the Special Projects Office of the US Navy in 1958. It developed a technique for evaluating the performance of large development projects, which became known as PERT - Project Evaluation and Review Technique. Other variations of the same approach are known as the Critical Path Method (CPM) or Critical Path Analysis (CPA) [11, 12].

The heart of any PERT chart is a network of tasks needed to complete a project, showing the order in which the tasks need to be completed and the dependencies between them. This is represented graphically as shown in Fig. 1. The diagram consists of a number of circles, representing events within the development lifecycle, such as the start or completion of a task, and lines, which represent the tasks themselves. Each task is additionally labeled by its time duration. Thus the task between events 4 and 5 is planned to take 3 time units. The primary benefit is the identification of the critical path. The critical path = total time for activities on this path is greater than any other path through the network (delay in any task on the critical path leads to a delay in the project). Diagram Symbols illustrated in Fig. 2.

Project network complexity is often recognized way, but not completely understood by everyone. Boushaala [13] proposed a new measure of project complexity. The developed measure showed more sensitivity to the changes in the network data and give accurate quantified results in evaluating the project complexity where the critical activities, the critical paths, number of critical activities to the total number of project activities, the length of critical path, the resource types and their availability are considered.

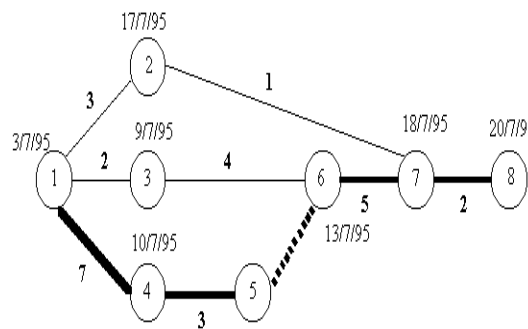


Fig. 1 Example of an activity network

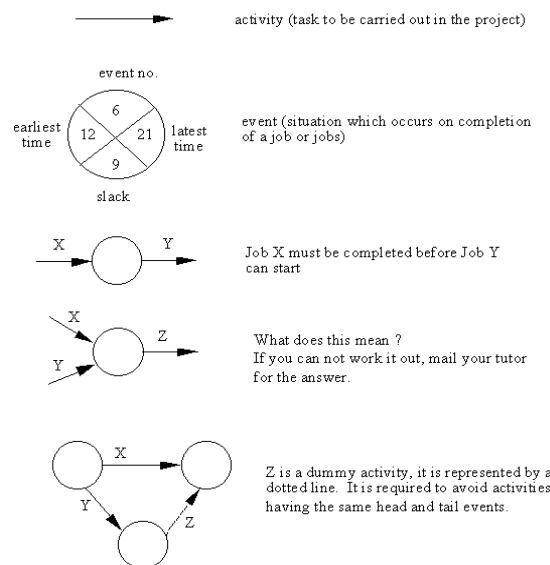


Fig. 2 Symbols used in activity networks

III. MODELING WITH PETRI NET'S

A Petri net (PN) is formally defined as a four-tuple $C = (P, T, I, O)$ where P is a finite set of places p , T is a finite set of transitions t , I is a mapping from transitions to a subset of places (bag of input places) such that $I(t)$ represents input places for transition t , and O is a mapping from transitions to a subset of places (a bag of output places) such that $O(t)$ represents output places for transition t . Multiple occurrences of each in the input and output bags are allowed [14, 15].

A PN can also be described by a bipartite directed graph with two types of nodes: circles for places and bars for transitions. Directed arcs connected places and transitions. Let $B(p, t)$ and $F(p, t)$ be, respectively, the number of occurrences of places P in the input and output bags of transition t . Then B , F and $D = F - B$, respectively, define the backward, forward and incidence matrices of the PN. These matrices define the topology of the PN. The dynamics of the PN are defined by marking μ of the PN; μ is a state vector with $\mu(p)$ is the number of tokens in place p . The dynamics of the PN are controlled by the execution of that PN. A PN executes by firing its transitions. A transition fires by removing tokens

from its input places and depositing tokens at its output places. A transition may fire if it is enabled. A transition t is enabled in marking μ if $\mu \geq B \cdot f_t$ where $f_t = (0, 0, \dots, 1, 0, \dots, 0)$ with 1 corresponding to transition t . If μ' is a new marking after firing transition t , then $\mu' = \mu + D \cdot f_t$ defines the dynamics of the PN. For a sequence σ of n transitions, the dynamics equation becomes $\mu_n = \mu_0 + D \cdot f$ where $f = \sum_{t \in \tau} f_t$, τ is a set of n transitions and μ_0 is the initial marking; f is called the firing vector of the sequence. Each marking defines a state. Firing a transition may result in a new state. All the possible states define the state space of the PN. From an analytical perspective, it is quite important to determine all the reachable states. It is also important to determine whether or not the PN is live or dead-lock free, bounded (number of tokens in any place is finite in any marking), conservative (the weighted number of tokens in any marking is fixed and finite) and consistent (there is a firing vector with all positive elements). A live and consistent PN is cyclic, which is typical property of manufacturing systems. One may also be interested in other features of a PN as a controller, such as recoverability and fairness. Some of these properties can be mathematically analysed through the P-and T-invariants of the PN [15, 16].

3.1 Marked Graph

A marked graph is a PN in which each place is an input for exactly one transition and an output for exactly one transition. Alternatively, we can say that each place exactly one input and one output [16, 17].

Definition: A marked graph is a PN $C = (P, T, I, O)$ such that for each $p_i \in P, |I(p_i)| = |\{t_j/p_i, O(t_j)\}| = 1$ and $|O(p_i)| = |\{t_j/p_i, I(t_j)\}| = 1$.

Marked graphs can model concurrence and synchronization but cannot model conflict or data-dependent decisions. The properties which have been investigated for marked graphs have been Liveness, safeness, and reach-ability. Marked graphs can model concurrence and synchronization but cannot model conflict or data-dependent decisions. The properties which have been investigated for marked graphs have been Liveness, safeness, and reach-ability. In the investigation of these properties, the major structural parts of a marked graph of interest are its cycles. A cycle in a marked graph is a sequence of transitions $t_{j_1}t_{j_2} \dots t_{j_k}$ such that for each t_{j_r} and $t_{j_{r+1}}$ in the sequence there is a place p_{ir} the $p_{ir} \in O(t_{j_r})$ and $p_{ir} \in I(t_{j_{r+1}})$ and $t_{j_1} = t_{j_k}$. A cycle is such a closed path from a transition back to that same transition.

If $P' \subset P$ is a subset of places that compose a cycle, then the characteristic vector $U = (u_i | i = \{1, 2, \dots, n\})$ such that $crd_i U = 1$ for $p_i \in P'$, is a p- invariant of PN. If U' and U'' are P- invariants of PN, then $U = U' + U'' + \dots$ is also a P-invariant of PN, where $crd_i U = u_i$ for $U = (u_1, u_2, \dots, u_i, \dots, u_r)$ [16].

3.2 P – Invariant

U is said to be a P-invariant of a PN if and only if $U = (u_1, u_2, \dots, u_n)$ is a vector such that $D \cdot U = 0$ and $u_i \geq 0$ for $i = 1, 2, \dots, n$ and D is the incidence matrix. The following theorem provides the condition allowing solving the reachability problem states. Given a Petri net $C = (P, T, I, O, \mu_0)$ with marking $\mu \in R(C, \mu_0)$ and marking $\mu' \in R(C, \mu_0)$. The problem is if μ' is reachable from μ , i.e. $\mu' \in R(C, \mu)$ [10,18].

Theorem (1): Let μ_0 be an initial marking and let $\mu \in R(C, \mu_0)$. If U is an P-invariant, then $\mu_0 U^T = \mu U^T$

The above theorem provides the condition allowing solving the so called reachability problem. The problem can be stated as following. Given a PN $C = (P, T, I, O, \mu_0)$ with marking $\mu \in R(C, \mu_0)$ and marking $\mu' \in R(C, \mu_0)$. Is μ' reachable from μ , i.e. $\mu' \in R(C, \mu)$? As shown in Fig. 3[19].

3.3 Software for a Functional Analysis of a Petri Net Model

The developed software enables analyzing of a system for its various properties irrespective of time. Functional analysis has been used in the Petri net to refer to this type of analysis. The software analysis is based on some approaches used to determine the various properties of PN model: liveness, safeness, reach-ability and conservativeness. Fig. 3 shows the flow chart of computing P-invariants.

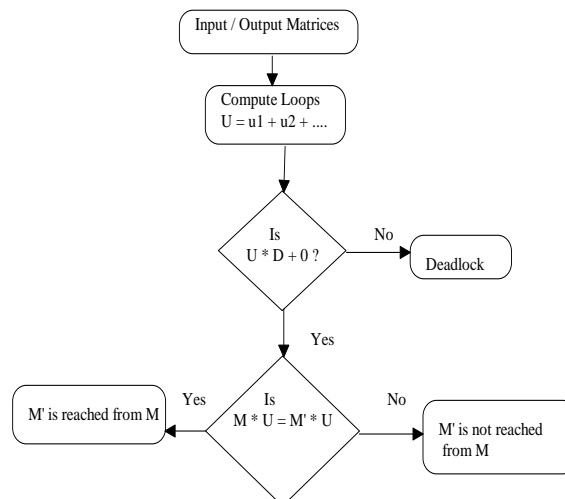


Fig. 3: Flow chart of computing P-invariants

IV. HEADINGS CONVERSION OF PERT/CPM CHARTS INTO EQUIVALENT PETRI NET MODEL

The Petri Nets show the same type of scheduling constraints as the PERT/CPM charts does. We can easily convert a PERT/CPM networks to a Petri Net model. Each activity in the PERT chart is represented by a place, while the precedence constraints are represented by the transitions. In CPM chart each event is represented by place (transition), while the activity is represented by transition (place). The Petri Net is excellent vehicle to represent the concurrency and precedence constraints of the PERT chart. In modelling the PERT chart with the Petri Net, activities that are represented by places and transitions are reserved to model multiple resources of limited amounts. Fig. 4 demonstrates the conversion process of PERT/CPM charts into Petri net chart.

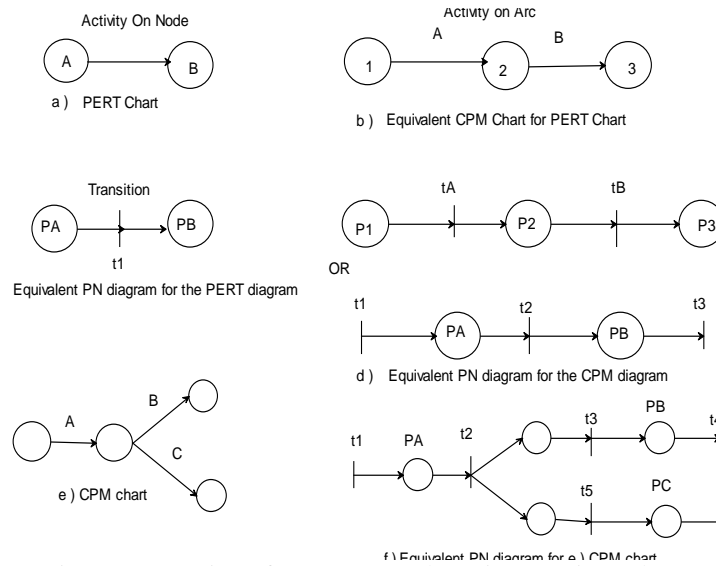


Fig. 4: Conversion of PERT/CPM charts into Petri net chart

- The Petri Nets are more powerful models as compared to the PERT/CPM charts. The following reasons may be given for it:
- The repeated performance of activities, if necessary, can be modelled by the PN.
 - Required resources per activity appear explicitly as tokens in representation.
 - Non-deterministic aspects can be dealt with. For example: the order in which a particular resource performs some tasks may not be totally specified.

V. MODELING PROJECT NETWORKS THROUGH PETRI NETS

By definition of a project, the project is completed when all the activities are processed exactly once. In PN terminology, this means that all the transitions must fire exactly once, i.e. $U \cdot D = 0$ for $U = 1$, where 1 is a column vector with all entries being one. This transition invariant verifies that the precedence relations among the activities are well established (the nodes are strongly connected), i.e. the project can be completed. Note that because the PN model of a project is a marked graph, the transition invariant is also a sufficient condition. $D \cdot 1 = 0$ means that the sum of each and every row of D should be zero. Hence, to verify that the project can be completed, it is sufficient to show that the incidence matrix of the PN model of the project has this property and that the PN is a marked graph. The following demonstrates this through an illustrative example. Consider a project consists of 8 activities A, B, C, D, E, F, G, and H with their times of completion as shown in Table 1 [18].

The project will be finished after 21 weeks, which is the CPM network contains two of critical paths. The first critical bath consists of 4 activities A, B, E, and H, and the second critical bath consists of 5 activities A, B, D1, F, and H. The activities D1 and D2 were dummies with zero duration. Fig. 5 shows A CPM project network for Table 1 with completion time for the project.

Table 1: The sequence of the project activities

Activity	Precedessor(s)	Duration (Weeks)
A	-----	3
B	A	6
C	A	4
D	A	3
E	B	4
F	B, C, D	4
G	B	1
H	E, F	8

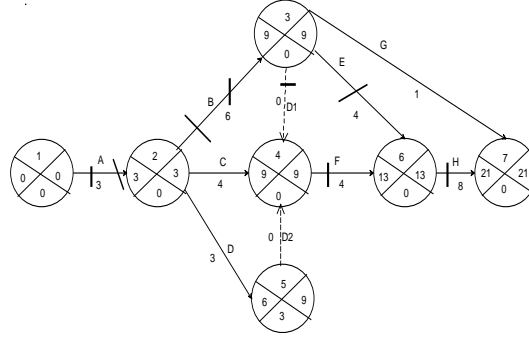


Fig. 5: A CPM project network for table 1 with completion time for the project [13]

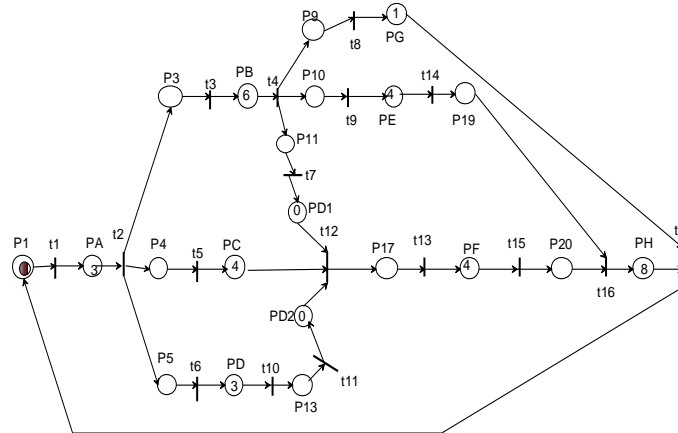


Fig. 6: Petri net (PN) model of figure 5

Fig. 6 shows the PN model of the project network in Fig. 5. For example activity A (PA) needs 3days to be completed. Activity times are denoted within the circles (places), which model the timed places. The bars denote immediate transitions, which model logical conditions (transitions t1 and t2 were pre and post conditions of PA) and take zero time to fire. The same as for the other activities from B through H including dummy activities D1 and D2 have pre and post conditions represented in transitions. All other places P1, P3, P4, ... , and P20 were connection places between all the project activities, for example P3 is the connection between the two activities PA and PB through t2 which is post condition of PA and t3 which is precondition of PB, as well as for the other remaining places (e.g. P4, P5, ... , P20) and needs zero time to be completed.

The incidence matrix of Fig. 6 is shown below. The sum of each and every row of the incidence matrix should be zero and each and every column should have at least one -1 and +1 entry to verify that the corresponding PN models the project correctly, i.e. all the activities, hence the project, can be completed

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16	t17			
p1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		
pA	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
p3	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
p4	0	1	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0		
p5	0	1	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0		
pB	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
pC	0	0	0	0	1	0	0	0	0	0	0	0	-1	0	0	0	0	0		
pD	0	0	0	0	0	1	0	0	0	-1	0	0	0	0	0	0	0	0		
p9	0	0	0	1	0	0	0	-1	0	0	0	0	0	0	0	0	0	0		
p10	0	0	0	1	0	0	0	0	-1	0	0	0	0	0	0	0	0	0		
p11	0	0	0	1	0	0	0	-1	0	0	0	0	0	0	0	0	0	0		
pD1	0	0	0	0	0	0	1	0	0	0	0	-1	0	0	0	0	0	0		
p13	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0		
pD2	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0		
pE	0	0	0	0	0	0	0	0	1	0	0	0	0	-1	0	0	0	0		
pG	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	-1		
p17	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0		
pF	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0		
p19	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0		
p20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0		
pH	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0		
-u ₁																		+ u ₁₇	= 0	
u ₁ - u ₂																			= 0	
																			u ₁₆ - u ₁₇	= 0

In order to illustrate the application of the approach proposed, let consider the example provided in the previous example of Fig. 6. An event graph (like PN in general) said to be strongly connected if there is a directed path joining any node A to any node B of the graph. The event graph presented in Fig. 6 strongly connected. We also define an elementary circuit in a strongly connected event as a directed path goes from one node, i.e. a place or transition, back to the same node, which any other node is not repeated. For instance, Fig. 6 exposes five elementary circuits, namely:

$$\gamma_1 = (p_1, t_1; p_A, t_2; p_3, t_3; p_B, t_4; p_9, t_8; p_G, t_{17};)$$

$$\gamma_2 = (p_1, t_1; p_A, t_2; p_3, t_3; p_B, t_4; p_{10}, t_9; p_E, t_{14}; p_{19}, t_{16}; p_H, t_{17};)$$

$$\gamma_3 = (p_1, t_1; p_A, t_2; p_3, t_3; p_B, t_4; p_{11}, t_7; p_{D1}, t_{12}; p_{17}, t_{13}; p_F, t_{15}; p_{20}, t_{16}; p_H, t_{17};)$$

$$\gamma_4 = (p_1, t_1; p_A, t_2; p_4, t_5; p_C, t_{12}; p_{17}, t_{13}; p_F, t_{15}; p_{20}, t_{16}; p_H, t_{17};)$$

$$\gamma_5 = (p_1, t_1; p_A, t_2; p_5, t_6; p_D, t_{10}; p_{13}, t_{11}; p_{D2}, t_{12}; p_{17}, t_{13}; p_F, t_{15}; p_{20}, t_{16}; p_H, t_{17};)$$

If the number of tokens in a marking remains fixed for all markings in the reachability set, the Petri net is then said to be strictly conservative. An immediate consequence of conservativeness is boundedness of the Petri net. The boundedness property implies that the number of tokens in any place does not increase beyond a limit. This in turn guarantees a finite reachability set for the Petri net. If there exist an U with all positive elements the Petri net is then said to be conservative.

If $U = (1, \dots, 1)$ then the Petri net is strictly conservative. The total number of tokens in γ_i ($i=1, 2, \dots, 5$) is then:

$$n_1(\gamma_1) = \mu(p_1) + \mu(p_A) + \mu(p_3) + \mu(p_B) + \mu(p_9) + \mu(p_E) + \mu(p_G) = 1 + 0 + 0 + 0 + 0 + 0 + 0 = 1$$

$$n_2(\gamma_2) = \mu(p_1) + \mu(p_A) + \mu(p_3) + \mu(p_B) + \mu(p_{10}) + \mu(p_G) + \mu(p_{19}) + \mu(p_H) = 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 1$$

$$n_3(\gamma_3) = \mu(p_1) + \mu(p_A) + \mu(p_3) + \mu(p_B) + \mu(p_{11}) + \mu(p_{D1}) + \mu(p_{17}) + \mu(p_F) + \mu(p_{20}) + \mu(p_H) = 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 1$$

$$n_4(\gamma_4) = \mu(p_1) + \mu(p_A) + \mu(p_4) + \mu(p_C) + \mu(p_{17}) + \mu(p_F) + \mu(p_{20}) + \mu(p_H) = 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 1$$

$$N_5(\gamma_5) = \mu(p_1) + \mu(p_A) + \mu(p_5) + \mu(p_D) + \mu(p_{13}) + \mu(p_{D2}) + \mu(p_{17}) + \mu(p_F) + \mu(p_{20}) + \mu(p_H) = 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 1$$

Therefore, the total P – invariants of all loops $U = (5, 5, 3, 1, 1, 3, 1, 1, 1, 1, 1, 1, 3, 3, 1, 3, 4)$.

By applying the equations : $U * D = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, this means that the system is deadlock free.

To compute the total time for each loop to find the critical path(s) as follows:

$$\gamma_1 = (p_1, p_A, p_3, p_B, p_9, p_G) = 0 + 3 + 0 + 6 + 0 + 1 = 10 \text{ weeks}$$

$$\gamma_2 = (p_1, p_A, p_3, p_B, p_{10}, p_E, p_{19}, p_H) = 0 + 3 + 6 + 0 + 4 + 0 + 8 = 21 \text{ weeks}$$

$$\gamma_3 = (p_1, p_A, p_3, p_B, p_{11}, p_{D1}, p_{17}, p_F, p_{20}, p_H) = 0 + 3 + 0 + 6 + 0 + 0 + 0 + 4 + 8 = 21 \text{ weeks}$$

$$\gamma_4 = (p_1, p_A, p_4, p_C, p_{17}, p_F, p_{20}, p_H) = 0 + 3 + 0 + 4 + 0 + 4 + 0 + 8 = 19 \text{ weeks}$$

$$\gamma_5 = (p_1, p_A, p_5, p_D, p_{13}, p_{D2}, p_{17}, p_F, p_{20}, p_H) = 0 + 3 + 0 + 3 + 0 + 0 + 4 + 0 + 8 = 18 \text{ weeks}$$

Then there is two critical paths $\underline{\gamma}_2$ and $\underline{\gamma}_3$, as the same as of the CPM network Figure 5.

VI. CONCLUSIONS AND REMARKS

The proposed PN based approach to project management facilitated modeling (resource-constrained) projects, and verifying some properties of the projects networks, exploiting the well known place (transition) invariants, through the basic PN semantics. That is, contrary to some approaches in the literature, there is no need to modify the basic PN semantics to model activities and decisions related to a project. This increases the capability of the basic PNs, without any extension in the theory, to model projects. Places were used to model all the preconditions, including resources, for an activity to start, which was modeled via a place (transition). The place (transition) invariants were then used to verify some properties of the projects. This PN model can then be analyzed through (PN based) scheduling techniques, used to find the critical path, used as a basis to develop algorithms for resource-constrained project management, and used for other decisions related to project management.

The transition invariant means that the sum of each and every row of the incidence matrix should be zero. Hence, to verify that the project can be completed, it is sufficient to show that the sum of each and every row of the incidence matrix is zero, and that each and every column has at least one -1 and +1 entry.

The proposed PN based approach enables the project manager to consider not only resources but also all different types of variables/constraints related to a project, e.g. costs of activities. Any of these variables can be considered fuzzy as well. In this case, fuzzy arc weights cause the firing rule to be modified. These are possible future research directions.

REFERENCES

- [1] B.S., Liang, J.N., Chen, and F.J., Wang, A project model for software development, *Journal of Information Science and Engineering*, 16, 2000, 423 – 446.
- [2] A.A., Desrochers, and R.Y., Al-Jaar, *Application of Petri nets in manufacturing systems: modeling, control, and performance analysis* (New York: IEEE Press, 1995).
- [3] V.A., Jeetendra, O.V., Chetty, and J., Reddy, Petri nets for project management and resource leveling, *International Journal of Advanced Manufacturing Technology*, 16, 2000, 516-520.
- [4] C.K., Chang and M., Christensen, A net practice for software project management, *IEEE Software*, November/ December, 1999, 80-88.
- [5] V.K., Kumar, and L.S., Ganesh, Use of Petri nets for resource allocation in projects, *IEEE Transactions on Engineering Management*, 45(1), 1998, 49 – 56.
- [6] W.M.P., Van der Aalst, Petri net based scheduling, *OR Spectrum*, 18, 1996, 219-229.

- [7] W., Herroelen, and R., Leus, Project scheduling under uncertainty: Survey and research potentials, *European Journal of Operational Research*, 165, 2005, 289-306.
- [8] S.M., Chen, and T.H., Chang, Finding multiple possible critical paths using fuzzy PERT, *IEEE Transactions on System Management, and Cybernetics Part B: Cybernetics*, 31 (6), 2001, 930 – 937.
- [9] K.W., Kim, Y.S., Yun, J.M., Yoon, M., Gen, and G., Yamazaki, Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling, *Computers in Industry*, 56, 2005, 143- 160.
- [10] V., Valls, F., Ballestín, and S., Quintanilla, Justification and RCPSP: A technique that pays, *European Journal of Operational Research*, 165, 2005, 375-386.
- [11] Erik and Will, *Project scheduling a research handbook*, (Kulwer Academic Publishers : New York, Boston, London, Moscow, and Dordrecht, 2002).
- [12] J.P., Lewis, *Project planning, scheduling and control*, (Mc Graw – Hill Companies, Inc., 2001).
- [13] A. M., Boushaala, Project complexity indices based on topology features, *Al-satil Journal*, 4 (8), 2010, 137-150.
- [14] T., Peterson, *Petri Net Theory and the Modeling of Systems* (Prentice Hall, 1981).
- [15] Z., Banaszak and K., Jedrzejek, Rule – based knowledge verification using Petri nets, 3rd *Turkish Symposium on AI and Networks*, June 1994.
- [16] P.M., del Foyo and J. R., Silva, Using timed Petri nets for modeling and verification of timed constrained workflow systems, *ABC Symposium Series in Mechatronics*, 3, 2008, 471-478.
- [17] Y., Chen, P., Fsu, and Y., Chang, A Petri net approach to support resource assignment in project management, *IEEE Transactions on Engineering Management*, 38(3), 2008, 564 – 574.
- [18] I. Amiruddin, K. Abdul Rashid, and J. Wisam, Literature review in resource – constrain critical path method technique, *European Journal of Scientific research*, 29(2), 2009, 222-236.
- [19] I. M., Buseif, *An approach to FMS design using SADT and PN tools*, doctoral diss., Faculty of Production Engineering, Warsaw University of Technology, Warsaw – Poland, 1997.