

Modelling and Analyzing a Hospital Procedure using a Petri-Net Approach

Mourtou Efstratia, Abdel-Badeeh M. Salem, and Pavlidis George

Abstract—Hierarchical high-level PNs (HHPNs) with time versions are a useful tool to model systems in a variety of application domains, ranging from logistics to complex workflows. This paper addresses an application domain which is receiving more and more attention: procedure that arranges the final inpatient charge in payment's office and their management. We shall prove that Petri net based analysis is able to improve the delays during the procedure, in order that inpatient charges could be more reliable and on time.

Keywords—eHealth, Petri-Nets, Hospital Services, Inpatient Charges, Workflow Modeling.

I. INTRODUCTION

GREEK Public Hospitals (GPH) are facing many problems in management and modernization of health services, in order to overcome the low degree of satisfaction of consumers. Manually produced, paper medical records are saturated with misfiled records, inaccessible data, illegible notes, and incomplete charts. In addition, hospital charges usually remain fuzzy and misty, while health Insurance Companies are deep in debt and crisis. Inpatients charges are not just prices to be paid, but also explanation of where those prices come from and why they are different from other hospitals. Thus, a different way of providing inpatients charges is needed in order to connect prices with the actual cost of providing hospital services [1]. On the other hand, information technology is becoming increasingly important, taking a leading role at many levels of the Greek National Health System (GNHS). Thus, hospitals have to change the paper-procedures in an electronically way, but a great concern of the rearrangement of operational processes is required, which focuses on re-definition of work flow and organisational structure. To tackle these problems we propose an approach based on Hierarchical high-level PNs (HHPNs) with time versions [2]. In this paper HHPNs are used to model the workflow of the required documents to arrange the final inpatient charges. Considering this workflow as a procedure

comprising of many tasks [3], we shall prove that we are able to improve the delays, in order that inpatient charges are reliable and on time. This paper is organised as follows: section (II) describes an overview of Petri nets approach, (III) gives the workflow definitions, (IV) explains a real case study, (V) discusses the evaluation of performance, and (VI) presents the conclusions and future work.

II. OVERVIEW OF PETRI-NET APPROACH

In the following, we give a short overview of Petri nets. More formal definition are given in [4-6].

Petri nets are bipartite graphs (for an introduction to graph theory [7]). That means there are two kinds of nodes in the net, called places and transitions, and arcs connect only two nodes of different types. While the places typically stand for the passive system elements which may be interpreted as conditions, the transitions represent the active system elements as events. Thus, every arc in the net connects an active and a passive element and since there are only direct arcs in Petri nets, they are represented by arrows. In graphical representation, places are depicted as circles and transitions as rectangles. According to the arc direction, every place has a set of pretransitions and a set of posttransitions. Since the presented net is transition-bordered, these sets are non-empty. In turn, transitions without replaces (post places) are called input (output) transition and are depicted as flat rectangles. The firing of input transition is not restricted from the net, i.e., the even modeled by an input transition may take places represents an events, which is only enabled to take place when all the preconditions of that event are fulfilled. The fulfillment of a condition is realized via tokens residing in places. If all preplaces of a transition are marked sufficiently with tokens (i.g., if the corresponding biochemical components are available), this transition may fire (i.g., the reaction takes place). If a transition fires, one token is removed from each of its places and added to each of its postplaces. Thus, the tokens are the dynamic elements of the system. Altogether, arcs connect an event with its preconditions which most must be fulfilled to trigger this event, and with its postconditions which will be fulfilled when the event takes place. Principally, a place in discrete net may carry any integer number of tokens, indicating different degrees of fulfillment, since the number of preplaces of a transition generally has not to be equal to the number of postplaces of this transition, the number of tokens in the whole net is not conserved. If a condition must be fulfilled, but the firing of a connected transition does not remove any tokens from it, these nodes are connected via two converse arcs. In the following, these arcs are represented by

Mourtou Efstratia is with Department of Business Administration of Patras University, Greece (corresponding author to provide phone: +30 2610 227847; fax: +30 2610 270469; e-mail: mourtou@agandreashosp.gr).

Abdel-Badeeh M. Salem is with the Computer Science Department, Faculty of Computer and Information sciences, Ain Shams University, Abbasia, Cairo-Egypt (e-mail: abmsalem@yahoo.com).

Pavlidis George is with the Department of Business Administration of Patras University, Greece (e-mail: (pvlds01@bma.upatras.gr).

bidirectional arrows as a short-hand notation, and are called read arcs. Fig.1 shows an example of Petri net, which illustrate the firing process. With the firing of input transition t_1 one token is put at place p_1 . this token enables the transitions t_2 and t_3 to fire, i.e., both of them are able to fire but only one of them can really fire, if there is only one token at their common replace (i.e. p_1). Since t_2 is an output transition, it can only remove this token from p_1 . The firing of t_3 would remove this token and put one token at each place p_2 and p_3 . A token from place p_3 is removed by firing of its posttransition t_4 . The distribution of the tokens over all places describes a certain system state and is called the marking of the net. According to this, the initial marking of a net describes the system state before any transitions have fired.

To avoid immoderate arc crossings, the construct of logical nodes is used in the graphical representation of the net. A logical node is represented by its name exists in multiple copies in the net, which are logically identical. Thus, a node defined as a logical one is identically equal to every other logical node with the same name and they are identified as one node which occurs in several graphical copies in the net. To distinguish these special nodes from ordinary ones, they are tinted grey. The model is one connected Petri net, but in the following it is presented and discussed in the form of three subnets which are connected via logical nodes in the original model.

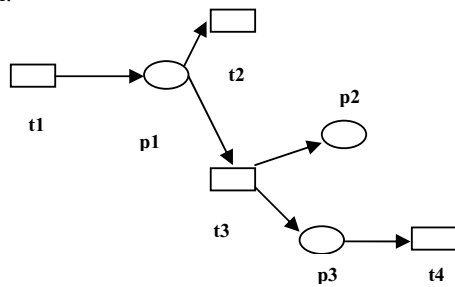


Fig. 1 Petri Net Concepts

III. WORKFLOW DEFINITIONS

In this section some terms of workflow modeling are introduced [8]:

Petri PN: A PN is defined as an ordered set $PN = (P, T, I, O)$ where:

P is the set of network places p_1, p_2, \dots, p_n , T is the set of network transitions t_1, t_2, \dots, t_n , I is the input function ($I: P \times T \rightarrow N$) that defines the set of oriented arcs from any place to any transition and O is the output function ($O: P \times T \rightarrow N$) that defines the set of oriented arcs from any transition to any place. For both functions, N takes values $0, 1, 2, \dots$

Marking: A PN marking is defined as the function $M: P \rightarrow N$ that assigns a non-zero integer in every PN place. Any marking can be represented as an n -dimensional vector whose elements are one-to-one corresponded in PN places.

The marking vector: This vector depicts the situation of the PN, namely the allocation of the marks in the places. The situation of the PN changes when the allocation of the marks is changing. Marks in any place may denote the number of resources, or they may indicate whether a condition is true, or

they may notify whether a job is operating (Job token). We suppose that any transition t_i activates only when for any marking of the place p , stands that $M(p) \geq I(p, t)$, $\forall p \in P$, while this transition fires and generates another marking M' , for which $M'(p) = M(p) - I(p, t) + O(p, t)$, $\forall p \in P$.

Task: As a task we consider any work to be done by one or more resources in a determined time interval.

Resources: Resources could be either human or material (e.g. computational systems) and they are handled by one or more resource managers. Likewise, any resource that executes a job is exclusively dedicated in this job.

Documentation: Documentation is considered in input or output of any task.

Procedure: A procedure is considered as a set of tasks, resources and control activities, which designate the routing and the synchronization of any task.

Job: A job is considered as a path that represents a predefined work according to a given procedure. Any job comprises any information about its relevant tasks, its documentation as well as the necessary resources. Any job inquires a procedure, while one procedure could be performed for different jobs

Workflow: A workflow is considered as a predefined set of jobs.

IV. CASE STUDY

The case study adopted here derives from a research in 41 GPH, which has the following results [9]:

(1). Only 39,02% of the hospitals use text-based logistics applications, though no interoperability with the operation theatre or the Laboratory Information System exists.

(2). Only 3% of the hospitals use computerized order entry for Medical and Surgical Equipment and Supply (MSES) for inpatients. Thereby, delays and fuzzy inpatients charges is a common problem in payment's office. This is not a surprise given the fact that there is a great absence of strong communication channels and of close collaboration between hospital departments and a low implementation of Greek Information Society [10].

A. Petri-Net Mapping Representation

The representation of the model is provided via the example of the following procedure. Consider an in-patient who was hospitalized for an interval time in a clinic. The in-patient office has to charge the patient or his insurance via a procedure **d**, which consists of the following elements (see Fig. 2):

- Three control activities **e₁, e₂, and e₃**, which are considered as transitions. These activities are required for the routing and the synchronization of the tasks as well as for the duplicating of the job tokens.
- Five tasks **t₁, t₂, t₃, t₄, and t₅**, which are considered as transitions.
- An input place **p₁** that represents the beginning of the procedure (**p₁ = begin_d**)
- A Resource Manager (RM), which includes the places request_resource (output), assigned_resource (input) and release_resource (output).

- An output place that represents the end of the procedure ($p_{11} = \text{end_d}$)
- Nine places p_2, p_3, \dots, p_{10} where every one is relating with a specific type that defines the form of the inclusive data.

Consider the control activity e_1 , which represents the search for the certificates of in-patient admission and discharges. This activity triggers two tasks t_1, t_2 which respectively correspond with the charge for medication and the charge for the MSES. When each of the above transitions activates, a request is sent at the RM in the place request_resource . Finally each transition is triggered by sending a token to the place release_resource (output) of the RM, while the final transition returns a job token to the procedure that started up the task. Each place of the RM belongs in different groups, depending on the corresponding task. For example, the place assigned_resource (input) for the task t_1 comprises the token that has the value 'Application_A31', while the same place for the task t_2 comprises the token that has the value 'MSES_B31'. When both tasks are completed, the control activity e_2 is triggered and so converting in an active situation, it duplicates the input token. Thereby two more tasks are triggered: t_3 that corresponds with the charge of medical tests and t_4 that corresponds with the charge of surgeries.

When both tasks are terminated, the control activity e_3 is triggered and converted in an active situation. Finally the transition t_5 , which includes information about the overall charge, is triggered. Suppose the input place p_1 includes a job token that triggers the control activity e_1 . Thereby this job token is duplicated and tasks t_1 and t_2 are triggered. The value of the job token is composed of an identifier, in order to be recognized by control activities and some references to documents. More analytically, the token that refers to task t_1 takes a value which includes the fields 'farm: A31' and 'report_farm' and the token that refers to task t_2 takes a value which includes the fields 'supply: B31' and 'report_supply'. Each 'report_%%' corresponds with the documentation of medication and MSES that were wasted for patient's hospitalization. Respectively, the token that refers to task t_3 takes a value which includes the fields 'test: A31' and 'report_test', while the token referring to task t_4 takes a value which includes the fields 'surgery: B31' and 'report_surgery'. Each of the above 'report_%%' corresponds with the documentation of the medical tests and the surgery during the patient's hospitalization.

B. Mapping Representation Using Time Colored Petri-Nets

Intending to improve the performance of the procedure, namely to increase the expedition of d, we shall add the concept of time into the aforementioned PN referring to Timed Colored Petri nets (TCPNs) [11],[12]. Attaching delays to transitions could give them the ability to model time and performance aspects can be investigated. Each of the aforementioned tokens receives time extension, namely a time value that represents the minimum duration that this token could be used. The key concept of the time assessment into procedure d is the following: To activate a task that requires t time units, the corresponding transition must set up for the output tokens more than t time units. In addition a global clock is introduced to the PN, whose values are integers and

represent model time. To each of the transitions a time stamp is added that represents the demanded interval for each task to be completed. More analytically, we add a new place 'wait' in transition e_1 , which represents for how long t_1 has to wait before it is triggered. Suppose the place 'wait' gets the value 50, namely 50 time units are required before t_1 is performed for one more time. We add to the transition e_1 the time stamp ($\&+10$) in order that tokens of the place p_2 are taking the time stamp ($t + 10$), where t is the time of e_1 acting. At the same time the arc that shows at the place p_2 adds one more delay to the token of p_2 , giving it the time stamp ($t + 10 + 50$). Likewise the token of p_2 is marked with the time stamp ($t + 60$). If we suppose that the initial value of t is zero, then the initials values of tokens p_2 and p_3 is 60. Each of the transitions t_1 and t_2 has a time stamp [$\&+\text{delay}()$], where $\text{delay}()$ is considered as a function that reverts an integer into the interval $[a, b]$. Under this consideration, any of the above transitions gets a variable duration, which in our model is representing the delay in searching the documentation of medication and MSES that were wasted for patient's hospitalization. Suppose the value of this delay belongs to the interval $[30, 80]$. After transitions t_1 and t_2 are triggered, each

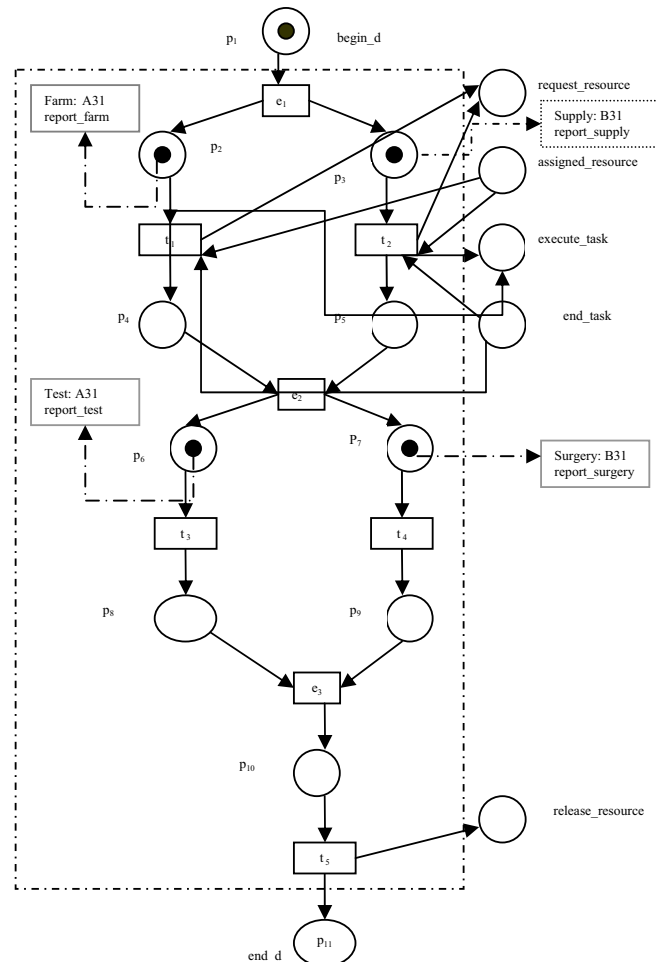


Fig. 2 Mapping procedure d

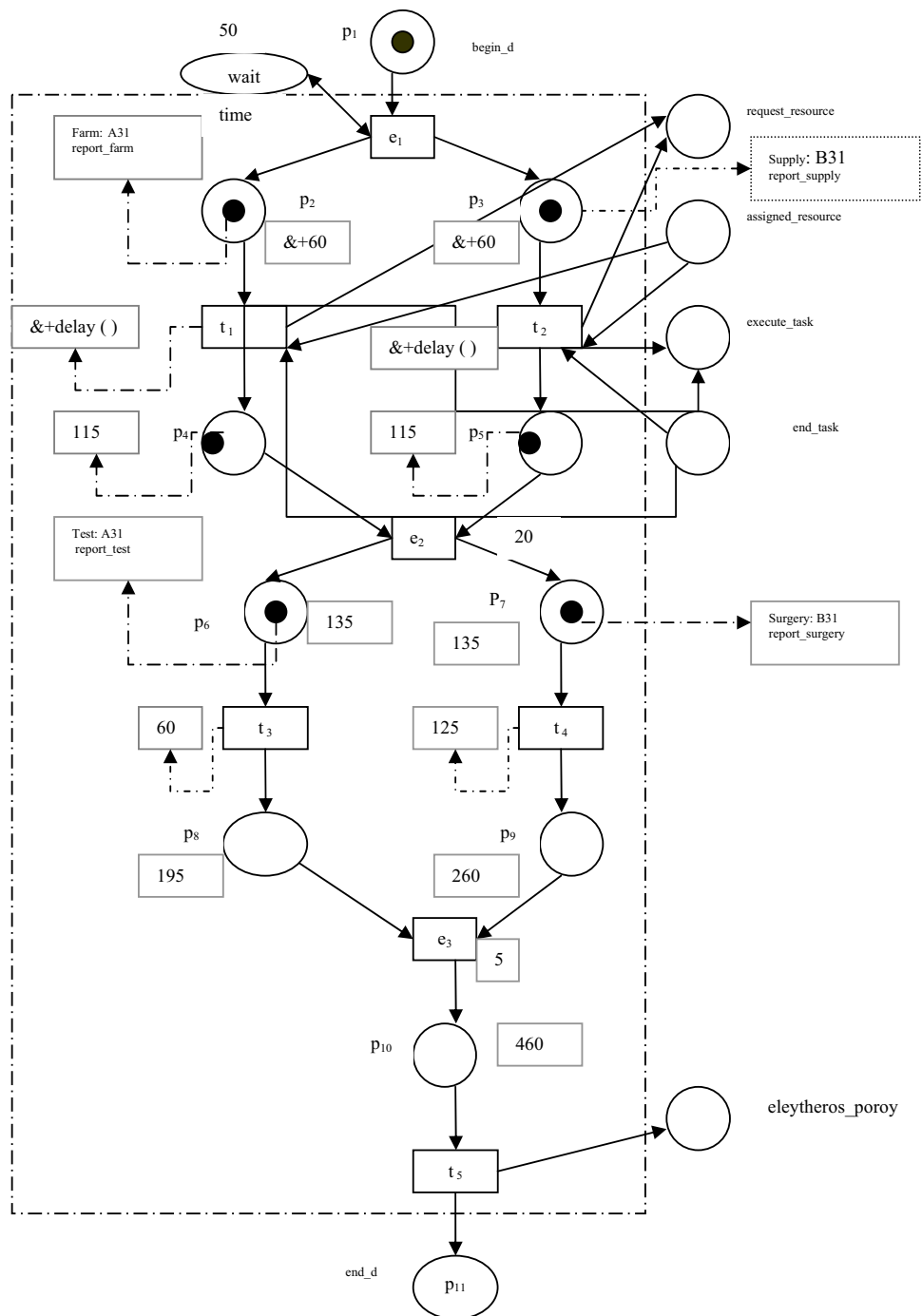


Fig. 3 Mapping procedure d, in a TCPN

of the tokens in places p_4 and p_5 receive the time stamp $(t + 60 + \text{delay}())$. Suppose that the value of $\text{delay}()$ is 55, namely the average of the time values that belong to interval $[30, 80]$. Then the time stamp of the places p_4 and p_5 are $60 + 55 = 115$. Consider that transition e_2 owns a stable time stamp whose value is 20. Then the tokens in places p_6 and p_7 receive the time stamp $115 + 20 = 135$.

Suppose that each of the transitions t_3 and t_4 has a time stamp $[\& + \text{delay}()]$, which in our model is representing the delay in searching the documentation of surgeries and medicals test that were wasted for patient's hospitalization. The first one is taking values into the interval $[50, 200]$ and the second into the interval $[30, 90]$. Due to synchronization problems of administrative and nursing services, we assume the bigger range of the first interval. Suppose that the value of $\text{delay}()$ of transitions t_3 and t_4 is respectively 60 and 125, namely the average of the time values of the corresponding intervals. Then the time stamps of the places p_8 and p_9 are

respectively $135 + 60 = 195$ and $135 + 125 = 260$.

Consider that transition e_3 owns a stable time stamp whose value is 5. Then the token in place p_{10} receives the time stamp $195 + 260 + 5 = 460$, while transition t_5 is considered without having time stamp, since it is always available. We also assume that transition t_5 owns a data type token, not coloured with time. Figure 3 depicts the procedure d via TCPNs.

V. THE EVALUATION OF PERFORMANCE

The performance of the procedure d could be improved if the delays of transitions either t_4 , or both t_1 and t_2 would be reduced. Suppose an on-line update of the payment's office by the patient's medication and MSES invokes. In that case the value of delay of transitions t_1 and t_2 is zero, and therefore the time stamp of each of the place p_4 and p_5 is 60. Thereby tokens in places p_6 and p_7 receive the time stamp $60 + 20 = 80$, returning a 40, 74% reduce in delays. Continuously, the token in place p_{10} receives the time stamp $140 + 205 + 5 = 350$, thereby the delay of the final charge is 23,91% reduced.

Let us assume that a better coordination between administrative and nursing service existed in order that the documents of surgeries would be sent in a faster way. At that case the value of $\text{delay}()$ of transition t_4 will be reduced, being less or equal with 60, which is the average of the interval $[30, 90]$. Thereby the time stamp of place p_9 will change into $135 + 60 = 195$, returning a 28,84% reduction in delay of task p_9 . Continuously, the token in place p_{10} receives the time stamp $195 + 195 + 5 = 395$, reducing the procedure of the final charge per 14,13%.

Suppose there was on line registration of the surgeries. Then the $\text{delay}()$ of transition t_4 would be equal to zero and the time stamp of place p_9 would be equal to 135, returning a 48,07% reduction of the task p_9 . Subsequently, the token in place p_{10} receives the time stamp $195 + 135 + 5 = 335$, resulting a 27,17% reduction of the delay of the procedure.

In addition, smaller values of the variable wait, namely less time in searching for the documentation of in-patient admission and discharges, could increase the possibility of time reduction during the procedure. If we consider the control transitions as control methods for every task, then possible

changes in the token types could dramatically change the performed tasks [13].

VI. CONCLUSION AND FUTURE WORK

In this paper we presented an approach to model the inpatient's charging procedure in terms of high-level Petri nets. To demonstrate the effectiveness of the introduced methodology, a case study was conducted and satisfactory results have been observed in the systems integration and the performance. Nevertheless, there remains a lot of work to be done.

This model can be used as a part of larger HHPNs with time versions, which could be comprised of smaller TCPNs that could describe in details the workflow of medical departments, connecting places, arcs and transitions. Thereby procedures that seem to be time consuming, become shorter, while the staff could know better the paths of their work and answer specific questions. In addition, managers would have the ability to discover and reform any weak points of persons or machines, during any procedure. At the same time, patients deserve to know how much their medical care will cost, and hospitals have an obligation to provide the transparent information. Likewise, this push toward transparency may pose challenges for hospitals to provide meaningful information to consumers and to Insurance Companies and give them the opportunity to understand the difficult economics of running a hospital.

REFERENCES

- [1] A. Camurri, P. Franchi, An approach to the design and implementation of the hierarchical control system of FMS, combining structured knowledge representation formalisms and high-level Petri nets, Proceedings of the IEEE International Conference on Robotics and Automation, Cincinnati, OH, USA, 13-18 May 1990, 1, pp. 520 – 525.
- [2] E. Mourtu, and G. Pavlidis, "Coding of Medical and Surgical Equipment and Supplies in Pharmacies of Greek Public Hospitals". Proceedings of the 10th International Symposium on Health Information Management - iSHIMR 2005, Thessaloniki, Greece: South East European Centre, pp 150-159.
- [3] H.J Genrich and K.Lautenbach, System Modelling with high level Petri net, *Theor. Comp Sci.* 13 (1981), pp. 109-136 MathSciNet.
- [4] B. Baumgarten, Petri nets basics and applications (in German). 2nd ed. Berlin: Spektrum Akademischer Verlag; 1996.
- [5] W. Reisig, Petri nets an introduction Heidelberg:Spring-Verlag; 1985.
- [6] T. Murata, Petri nets: properties, analysis and applications, *Proc. IEEE* 77 (1989) (4) pp. 541-579.
- [7] B. Bollabas, Modern graph theory. New York: Spring-Verlag, 1998.
- [8] Muller R, Rahm E. Rule-Based Dynamic Modification of Workflows in a Medical Domain. Proceedings of BTW99, Freiburg in Breisgau:Springer, 1999, p. 429-448.
- [9] E. Mourtu, The role of Rules and Regulations in the management of technological change in hospitals. *The journal on Information Technology in Healthcare*, 2006, 4(3), 147-154.
- [10] Greek Computer Society (EPY), Thesis of EPY about Information Society, Vol. 92, January-March 2003.
- [11] Gerogiannis, V., Kameas, A., and Pintelas, P. Comparative study and categorization of high-level petri nets. *Journal of Systems and Software*, 1998, 43(2), 133-160.
- [12] K. Jensen, Coloured Petri nets. Basic concepts, analysis methods and practical use, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, Berlin (1992).
- [13] Kurt, J., Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin Heidelberg New York, 1992, 1.