Modified Petri Nets for Hybrid Systems Modeling - a Case Study

Mircea-Adrian Drighiciu, Daniel Cristian Cismaru
University of Craiova / Faculty of Electrical Engineering, Craiova, Romania, adrighiciu@em.ucv.ro

Abstract - **In this paper we deal with the problem of modeling and simulation a DC electric drive system using a Hybrid Petri Net formalism, considering that the electric drive is a hybrid system which switches between its states under the occurrence of generic and non-generic external events. The synthesized topology is called a Modified Hybrid Petri Net and is a hybrid model that combines both differentials and discrete Petri Nets dynamic and enables us to simulate and to analyzing the behavior of DC electrical drives in different operating regimes due to the particularities imposed by the load of electrical motor. A real application case was finally considered; different scenarios were analyzed, depending on the nature of the engine load and the specific working conditions. The correctness of the initial model was verified by simulation and, also, by comparison of the results with those obtained experimentally, on a physical model. The modified Petri Nets model is carried out with an available software Visual Object Net ++, which allowed at the same time the synthesis of a hierarchical Object Petri Net structure. All models obtained through this method had been synthesized through an ascendant technique, around a central core, by adding new elements according to the studied processes.**

Cuvinte cheie: *Sisteme Hibride, modelare, simulare, Rețele Petri, programare orientată pe obiect.*

Keywords: *Hybrid Systems, modeling, simulation, Petri Nets, objects-oriented methods.*

I. Introduction

The approaches related to the modeling, analysis and control of hybrid systems are, in the last years, issues of great interest within the research teams, due to the practical applications in the various fields. The main paradigm of all researches related to a hybrid system considers that such a system is most often, as a hierarchical structure. On a basic level, there are continuous sub-systems (processes), linear or nonlinear, having an open loop or closed loop dynamic, controlled by discrete event drive systems from a high level and driven by occurrence of asynchronous events [2], [3], [4], [5], [6], [7], [8], [9]. For the same set of events, the system reaches always the same states. The evolution of the system between its discrete switches is described such a continuous dynamic which can be represented by differential equation models

Starting from this paradigm, the paper will provide the implementation of a basic model for analysing the DC electric drives, assuming, from the start, that the system is a hybrid one. Modifying the dynamics of the system is made under the occurrence of a series of internal and external events. The internal events can be determined by the variation limits of a system's state variable, while the external events can be determined by the action of some disturbances or commands [6], [7], [8], [9], [11].

An adequate instrument for this means is the formalism of Petri Nets, utilized for shaping and analysing the distributed and concurrent systems, as well as for creating interesting practical applications. By owning a friendly graphic interface and an adequate mathematical formalism, Petri Nets allow qualitative and quantitative analysis of the shaped processes [2], [3], [4], [5], [14], [15], [16], [17], [18], [19].

II. Background on Hybrid Petri Nets

In this paper, the authors did not propose a detailed presentation of the basic definitions and concepts specific to this formalism. Their approach is focused on the specific representations of one particularly class of Hybrid Petri Nets, suitable for modeling of a particular class of hybrid processes, assuming that the main definitions and rules of this formalism are known [2], [3], [4], [5], [6], [8], [9], [10].

However, we must note here that among the main discrete event dynamic systems models, Petri Net techniques have a special position because they seem to be more accepted in industry and the applications domain, in general.

The first extension of Discrete Petri Nets (DPNs) was the Continuous Petri Nets (CPNs) in which the marked places are no more represented through integers but through real numbers. These types of structure can, therefore, shape continuous processes and systems. Subsequent have been defined Various timed CPNs models, corresponding to different calculation of the firing speeds associated to the transitions [7], [8], [15], [17], [18].

If combined Discrete Petri Nets and Continuous Petri Nets in one single model, it has resulted the Hybrid Petri Nets (HPNs) which have their dynamics, found in both continuous and discrete systems. So, a Hybrid Petri Net is a model made up of DPNs and CPNs that interact with each other. (Fig.1). A such topology can contain in its structure discrete, real and symbolic variables usually encountered in other models or representation of hybrid systems.

A model of this type inherits all the modeling facilities that the discrete and continuous Petri Nets contain and, in addition, allows the unitary treatment of the entire dynamics of the system. [3], [6], [7], [8], [11].

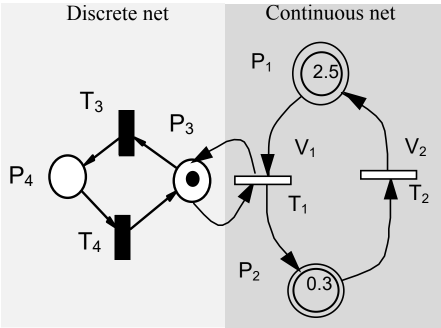

Fig.1 A Hybrid Petri net structure.

A hybrid modified Petri Net (Fig.1) is composed of two distinct categories of subnets: one *discrete* (P_3, P_4, T_3 and T_4) and the other *continuous* (P_1, P_2, T_1 and T_2). These are interconnected through several categories of arcs, depending on the mechanism by which an adequate dynamic of the whole network is ensured. In many cases, one part can influence the behavior of the other part, changing or no its own marking. In other situations, the firing of a discrete transition can modify both the discrete and the continuous marking of the model. Usually, the firing speeds – associated to the continuous transitions of the net – are constants and theirs values can be the same, or different in various scenarios. The main inconvenient of basic HPN models is that they cannot represent negative continuous variables. which sometimes correspond to the state variables of the process.

In order to overcome this shortcoming Differential Petri Nets (DPNs) were introduced. These are an extension of basic HPNs and can represent simultaneously continuous dynamic systems, modeled as systems of ordinary differential equations, and discrete event driven systems [9], [10], [15]. The novel features of DPNs are the negative real values accepted for place markings and the use of an integration mechanism for the approximate representation of the continuous systems. Under the assumption that the continuous system can be represented by a *n* linear first order difference state equation, they are powerful enough to model in a single graph a hybrid system. Also, similar to HPNs, a state equation can be formulated, from which it is possible to deduce from a initial marking (initial, or current state of the system) at time t_i the reachable marking (the state that can be reached in the evolution of the process) at the date t_k:

$$M(t_k) = M(t_i) + U \cdot \left(\sigma(t_k) + \int_{t_i}^{t_k} v(u)\,du\right) \qquad (1)$$

where $\boldsymbol{U}$ is the incidence matrix, the characteristic vector $\boldsymbol{\sigma}$(t) represents the firing sequence for the discrete transitions and the speeds vector $\boldsymbol{v}$(t) contains the instantaneous firing speeds of differential transitions.

Another feature that greatly increases the modeling power is given by the possibility of associating some of the model transitions with firing speeds represented by algebraic functional expression, whose arguments can be the marking quantities of arbitrary places of the net (Fig. 2), [11], [12], [13]. Also, it is possible that some places are linked by test arcs – which not change at all the marking of connected places - to the output transitions, authorizing in this way their firing. The result is named Modified Petri Nets – (MPNs).

In the figure below, the firing speed *v* is the speed of token flow from place P_1 into the place P_2 [9], [10]. The transition T_1 is active as long as the input place marking is greater than zero; it can be inactivated only by the empty discrete input places.

The test arc (dotted line represented) P_1 – T_1 does not allow token flow between P_1 and P_2.

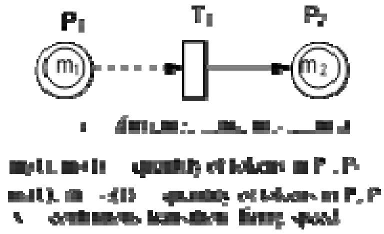

Fig.2 An MPN topology.

According to the above representation, the instantaneous firing speed – *v* – associated with the continuous transition T_1 linking two continuous places (P_i to P_{i+1}) is given by:

$$v = \frac{dm_{i+1}(t)}{dt} = -\frac{dm_i(t)}{dt} \qquad (2)$$

If the place Pi is connected through „ n” input arcs from the rest of PN model (Fig.3), then:

$$\dot{m}_i(t) = \frac{dm_i(t)}{dt} = \Sigma_j\, v_j \qquad (3)$$

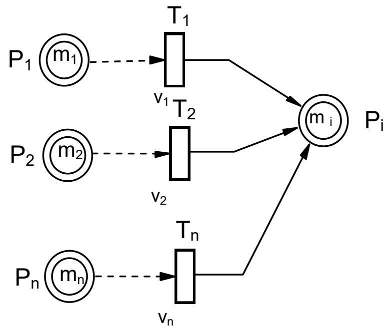

Fig.3 Continuous place with „ n” input transitions.

In order to check all of goods properties of the models, algebraic techniques can be used in addition with various software tools, which assures in the meantime different simulation scenarios in many different initial conditions [11], [12], [13].

Accuracy of synthesized models can be verified by comparing the simulation results with those obtained by using other alternative modeling techniques, using classical software tools (Matlab – Simulink, Modelica etc.) or with laboratory experimental test, using and starting from a physical model.

III. CASE STUDY

Using combined synthesis techniques, in order to highlight the ability of MPSs to represent hybrid systems, the authors proposed a basic model of a DC electric drive system, considered as a hybrid structure. Hence, starting from the basic physical principles of the system and thanks a linear and simplified mathematical model of this, for the beginning the basic topology of the Petri Net model was realized (called the parent Petri Net). This parent net was used to model the process in different operating scenarios, then this structure was developed by adding additional elements to illustrate different operating situations, in order to obtain a modular and flexible framework – a Object Petri Net (OPN).

A. Mathematical model

Usually, the initial mathematical model of an DC electric drive (Fig.4) consist of a linear differential equations system, which emphasizes the main system structure and stipulate the behavior rules for this one.

It is obvious that such topology is proper to describe a hybrid and variable system structure, represented by continuous models that switch successively depending on the specific operating conditions, and which are used successively (one or each other) for performances analysis. The electric drive can work in different continuous operating regimes, with the load permanently connected to the motor shaft, or in intermittent operating regimes, in which the load is periodically connected and disconnected. So, both the system structure and the mathematical model used for the analysis are variable, depending on the specific working conditions.

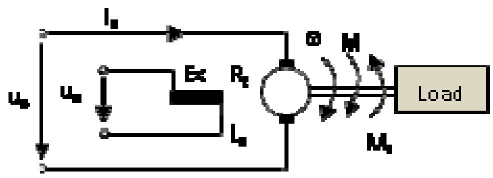

Fig.4 An DC motor drive structure

In a simplified mathematical model it's assumed that the inductivities of the armature circuits have constant values (L_a = ct., L_e = ct.), the undesirable effect of the reaction of the inducer, the voltage drop at engine brushes ($\Delta u_p = 0$) and the friction torques are neglected and, moreover, considering that the drive is controlled in the rotor (ϕ = ct.), the mathematical model can be represented in relative units [1], [12]:

$$u = \rho \cdot i + \rho \frac{T_a}{T_m} \frac{di}{d\tau} + \varphi \cdot v \tag{4}$$

$$\varphi \cdot i = \mu_s + \frac{dv}{d\tau} \tag{5}$$

The significance of the variables in the mathematical model described by (4) and (5) is the one generally accepted in the specialized literature [1], [11], [12].

Starting from this point, the basic idea used in the synthesis of the Petri Net model was to associate the input, state and output variables of the mathematical model with continuous places of an MPN model. Also, to the continuous transitions of the model will be associated variable firing speeds in accordance with (2) and (3) rules, having as result a modular and flexible model topology with a variable structure.

B. MPN model

Following the above idea, starting from (4) and (5), using the facilities offered by DPNs and MPNs formalism a first structure of the model was synthesized (Fig.5). In this model, the DC electric drive works with the load disconnected from the motor shaft (the load torque $\mu_s = 0$).

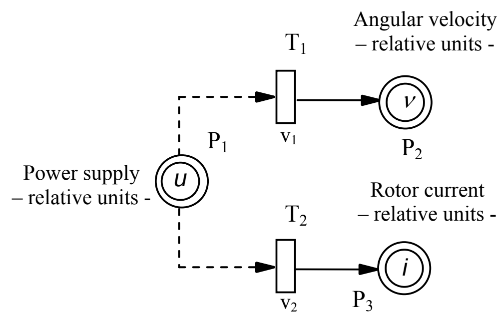

Fig.5 MPN model of an DC electrical drive for $\mu_s = 0$

The place P_1 contains the power supply variable (the input variable), expressed in relative units - *u*. Its marking (1 for rated value) authorizes the continuous firing, with speed v_1 and v_2 respectively of both continuous transitions T_1 and T_2. In accordance with evolution rules of MPNs, during continuous firing of these transitions, the marking of the place P_1 does not change. The P_2 place contains the instantaneous angular velocity variable (state/output variable), also expressed in relative units - *ν*. Similarly, in the place P_3 was expressed instantaneous variable of the current in the motor rotor (state variable) - *i*. The mathematical expressions of the two instantaneous and continuous firing speeds v_1 and v_2 respectively were obtained directly from (4) and (5), according (2) and (3) rules:

$$v_1 = \frac{dv}{d\tau} = \varphi \cdot i \tag{6}$$

$$v_2 = \frac{di}{d\tau} = -\frac{T_m}{T_a} \cdot i - \frac{T_m}{T_a} \cdot \frac{\varphi}{\rho} \cdot v + \frac{1}{\rho} \frac{T_m}{T_a} \cdot u \tag{7}$$

The parent net model (Fig.5) was synthesized using a specialized software tool, which allowed, at the same time, both the verification of its correctness and the analysis of the behavioral properties, via simulation [11], [12], [15]. Fig.6 illustrates the transient regime during the starting process with full rated voltage and under no-load condition, which essentially means that the load torque $\mu_s = 0$, for an DC electric drive with: P_N = 2,64 kW, U_N = 110 V, I_N = 30 A, n_N = 1500 rpm, R_a = 0,36 ohm, L_a = 0,125 H, J = 1,2 kgm^2. These results were obtained following the simulation scenario of MPN illustrated in Fig.5, which will be called M_1 – model.

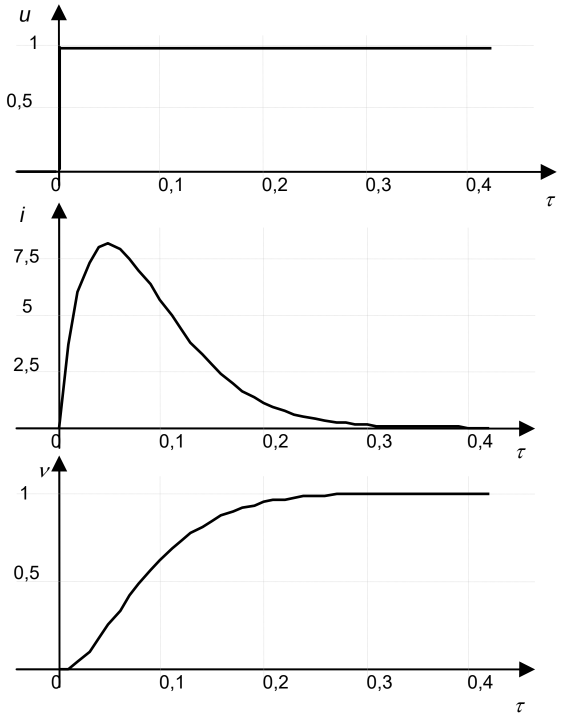

Fig.6 The transient regime during the starting process ($\mu_s = 0$) reached by simulating the MPN model of an DC electrical drive, using the VisualObjectNet++ tool.

As already stated, the basic topology of the MPN model (the parent net – Fig.5) denotes a modular structure, which can be enriched and refined by adding other different functional Petri Nets elements, in accordance with the real modeled system and its dynamics.

Hence, for illustrate the behavior of the DC motor in presence of the load torque Ms, a complementary sub – net can be added at the parent model (Fig.7). The new MPN model – M_2 model - achieved is also a modular and synthetic framework which contains in its structure the topology of the basic (parent) model M_1 (Fig.5). It allows simulations and analysis of behavioral properties of the electric drive under different initial conditions and various operating regimes (starting, speed control, electric breaking etc.) as well for different mathematical expressions of the load torque [11], [12], [13], [14].

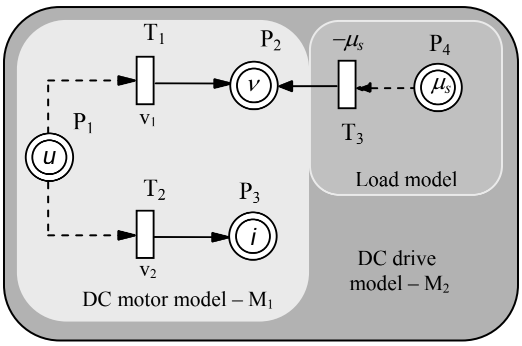

Fig.7 The MPN model of an DC electric drive for $\mu_s = 1$.

Each continuous transition of the model it's always active; these can only be disabled by a null marking of discrete input places. The test arcs $P_1 – T_1$, $P_1 – T_2$ and $P_4 – T_3$ does not allow token flow of their input places (P_1 respectively P_4), but only authorized the firing of continuous transition of the model. This makes it possible to model various categories of systems without feedback: the token quantity of P_1 and P_4 is not influenced.

The enriched M_2 model of the electric drive was synthesized and then its behavioral properties were verified by on-line simulation using Visual Object Net++ software tool [12], [13]. Also, the simulation results were compared with the laboratory experimental results or other sets of Matlab-Simulink models simulations, for the same scenarios and initial conditions. Fig. 8 illustrates the transient regime of the angular speed and of the rotor current respectively, during the starting process with full rated voltage of the power supply (the initial marking into P_1 is u = 1), and at rated load torque ($\mu_s = 1$). The results were obtained by simulating, for an DC electric drive with the same rated values as those used for the representation in the Fig.6.

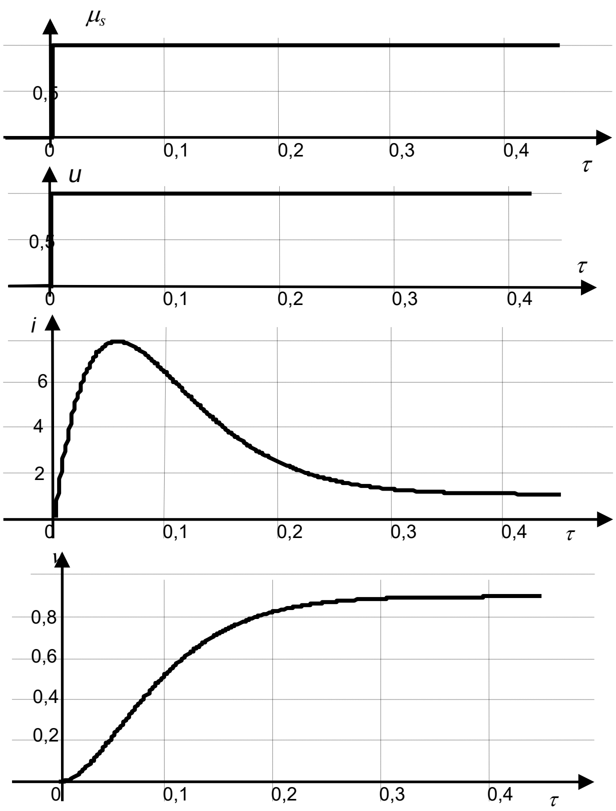

Fig. 8 On-line simulation results of the MPN model for $M_s = M_N$.

If the load of the motor is constant as value (M_s=ct.) and it's periodically connected and disconnected from the shaft, the electric drive works in an intermittent operating regime. In this case, the topology of the model can be expanded by adding a sub-model of the control structure. This sub-model can be a discrete, temporized Petri Net (Fig.9). Also, the commands for connecting/ disconnecting the load to the motor shaft can be given according to the values of state variables achieved from the process.

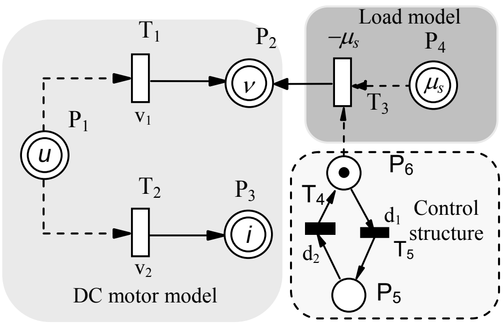

Fig.9 The Hybrid Petri net model of the electric DC drive working in an intermittent operating regime, with a constant load.

Dynamics of the whole system in the states space is determined by the occurrence of the external events (connect/disconnect commands of the load). This behavior ensures greater flexibility for the entire model, that switches between M_1 and M_2 topologies. In accordance with these external command-type events, the hybrid system reaches a new quality and its behavior can be represented as a three – dimensional reachability graph (Fig.10).

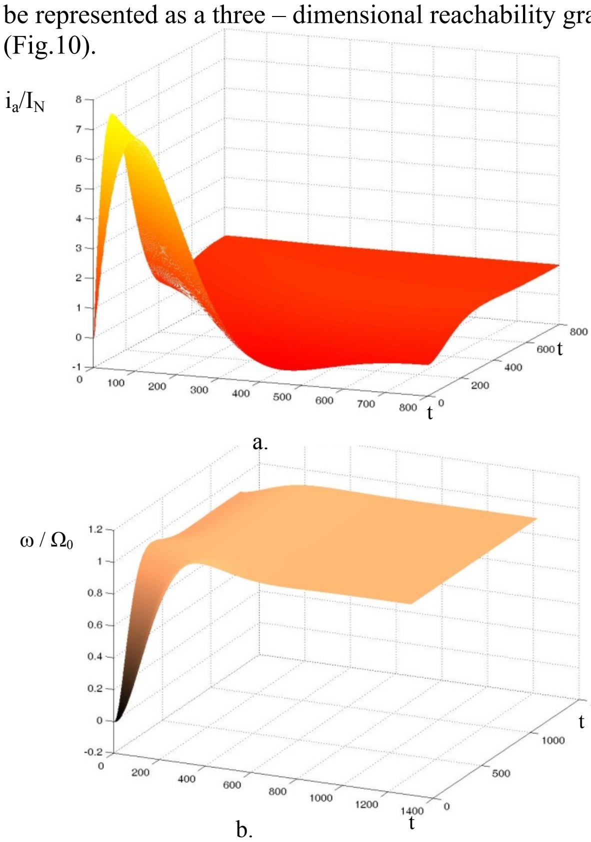

Fig.10 The reachability graphs of the hybrid system: a) rotor current; b) angular velocity.

These representations (named reachability graph) provides complete information about the whole set of values reached by the state variables of the system (the rotor current and the angular velocity in this case) as a result of occurrence of external event sequences.

Through minimal modifications of the initial MPN topology, different variants of the inner model can be achieved. These can be used to analyze particular situations and scenarios.

Thus, for a linear dependence between the load torque of the motor and its angular velocity ($\mu_s = k \cdot \nu$) the M_2 model (Fig.7) can be enriched by adding two discrete transitions (T_4 and T_5), which induces only minimal changes in the basic structure of the parent MPN (Fig.11). The discrete transition T_4 is permanently enabled and the weight of the T_4 – P_4 arc – $k \cdot \nu$ - assures a continuous variation of the P_4 marking.

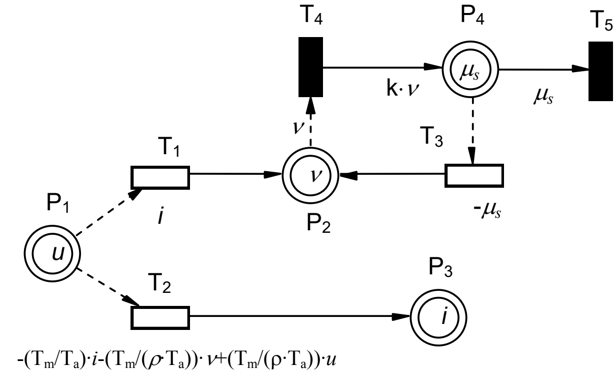

Fig.11 DC electric drive MPN model for $\mu_s = k \cdot \nu$.

C. *Object Petri Net paradigm*

The formalism of Petri Nets, discrete, continuous or hybrid is suitable for the analysis of small systems. In the case of large-scale systems or processes, the topology of the model and its own dynamics can raise serious problems, especially related to the complexity of the system and the "explosion" of the state space. Once the Petri Net model is built, the structural changes that can be made are, usually, small. A possible way to solve some of these drawbacks is to introduce the Hybrid Object Petri Nets (RPHOs) whose elements are Petri Nets sub-models, organized as independent entities, with their own structure and dynamics, located on different levels of organization and directly or through auxiliary elements of the same nature interconnected [11], [12], [13]. An important advantage of this mode of representation is the possibility of inheriting the properties of the old class by the new class achieved. Also, the objects belonging to it include the properties of the primary objects. The whole model of the system (process) is thus represented by the interaction of several object Petri Nets that communicate with the user through input/output interfaces. The structure of the model provides information on the organization and interaction of its "attributes", considered variables of the Petri Net model, related to its positions and described by an appropriate marking.

Using this concept, also the software tool facilities Fig.12 shows the Object Petri Net hierarchical structure of the electric drive model (Fig.7); the marking of published interface places can be modified from the environment, or by interaction between many object nets. Visual Object Net++ is a software suitable tool which provides such hierarchical structures [11], [12].

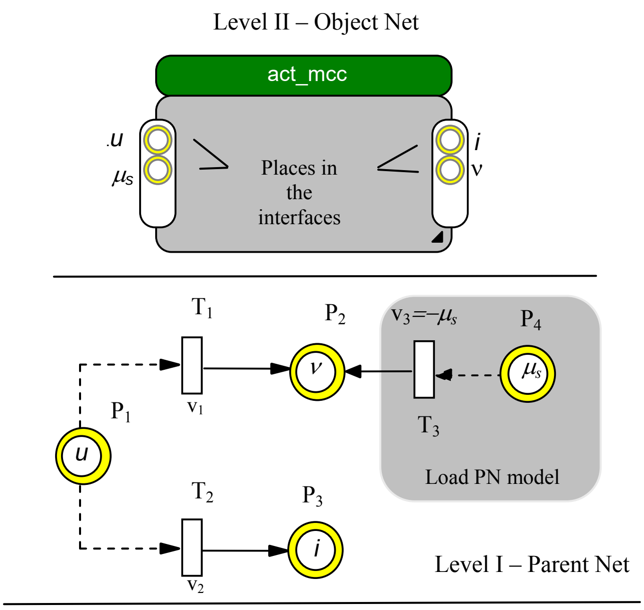

Fig.12 The Object Net built from the initial parent MPN model.

IV. CONCLUSIONS

The analysis has shown the possibility of applying the formalism of Petri Nets in a domain governed by abstract analytical models, adapted to the continuous aspect of the deployed processes. Aside from the classical representations, the Continuous Petri Nets model proves to be efficient in obtaining basic modular structures, which have a high abstraction capacity, organized afterward in network-type objects with increased flexibility and variable structure.

Moreover, the dynamic of the process governed by asynchronous external events had imposed the extension of the formalism to the Generalized Hybrid Petri Nets. In their structure had been used both discrete elements, for highlighting how the events act, as well as elements through which the operating restrictions limit the evolution of the continuous sizes like the test arcs.

The models obtained through this method had been synthesized through an ascendant technique, around a central core, by adding new elements according to the studied processes.

Especially useful for the analysis, the Visual Object Net ++ software tool had allowed the gradual construction of models and, at the same time, a simulation check-up of their properties. There are software tools which provide the user with a complete set of information, useful in drawing conclusions or come up with generalizing phrases, along the step by step evolution of the model's marking (state). No matter the type of software which is going to be used, it requires a good understanding of each network's own mechanisms and good knowledge of the physical system. The heuristic attribute of the models allow, almost every time, a thorough structuring of the initial topology, which simplifies the structure, and leads to more synthetic networks, in spite of the veiled correspondence with the shaped or analyzed physical model (especially high-level networks.)

The results received after the simulations on hybrid models (a mix between Continuous Nets and Discrete Nets) are more than encouraging and plead about extending this method to develop some command structures which should include Petri Nets Complier, code generators for command-control software tools.

ACKNOWLEDGMENT

This work was supported by research program of the Electrical Engineering Department, financed by the University of Craiova.

Contribution of authors:
First author – 80%
Coauthor – 20%

Received on July 17, 2019
Editorial Approval on November 20, 2019.

REFERENCES

[1] L. Cantemir, "Some principles problems concerning the electromechanical drive", Proceedings of the 9th National Conference on Electrical Drives, Craiova, 8-9 October, pp.13-19, 1998.

[2] Baniardalani, S., Modeling of Discrete-Time Systems Using Petri Nets, 2019 27th Iranian Conference on Electrical Engineering (ICEE), 30 April-2 May, Yazd, Iran, 2019.

[3] M. S. Branicky , "A Unified Framework for Hybrid Control□, Proc. 33rd Conference on Decision and Control, Lake Buena Vista, Fl., USA, pp. 4228 – 4234, 1994.

[4] F. Balduzzi, A. Giua, C. Seatzu, "Modelling and simulation of manufacturing systems using first-order hybrid petri nets", Int. J. Prod. Res. 39(2), pp.255–282, 2011.

[5] F. Capkovic, "Petri Nets in Discrete-Event and Hybrid Systems Modelling, Analysing, Performance Evaluation and Control", Automation 2017, Advances in Intelligent Systems and Computing 550, DOI 10.1007/978-3-319-54042-9 1, pp.3 – 21, 2017.

[6] R. David, H. Alla. Du Grafcet aux Réseaux de Petri, Hermes , Paris, 1992.

[7] R. David, H. Alla, "On Hybrid Petri Nets", Discrete Event Dynamic Systems: Theory and Applications, No.11, pp.9 – 40, 2001.

[8] R. David, H. Alla. Discrete, Continuous, and Hybrid Petri Nets, Springer- Verlag, Berlin, 2005.

[9] G. N. Davrazos, N. T. Koussoulas, "A General Methodology for Stability Analysis of Differential Petri Nets", Proceedings of the 10th Mediteraneean Conference on Control and Automation – MED2002, Lisbon, Portugal, July 9-12, 2002.

[10] Isabel Demongodin, N.T. Koussoulas, "Differential Petri Net Models for Industrial Automation and Supervisory Control", IEEE Transactions on Systems, Man and Cybernetics – Part C:Applications and Reviews, Vol.36, No.4, pp.543-553, 2006.

[11] R. Drath, "Hybrid Object Nets: An Object Oriented Concept for Modelling Complex Hybrid Systems", Dynamical Hybrid Systems, ADPM98, Reims, 1998.

[12] R. Drath. Tool Visual Object Net ++, Available: http:// www.daimi.au.dk/ PetriNets/tools, section "Visual Object Net++".

[13] M. A. Drighiciu, Gh. Manolea, D. C. Cismaru, Anca Petrişor, "Hybrid Petri Nets as a New Formalism for Modeling Electrical Drives", Proceedings of International Symposium on Power Electronics, Electrical Drives, Automation and Motion – SPEEDAM 2008, ISBN: 978-1-4244-1664-6, IEEE Catalog Number: CFP0848A-CDR, Ischia, Italy, 11th – 13th June, pp.626-631, 2008.

[14] M.A.Drighiciu, " Hybrid Systems Modeling Approach with Petri Nets", Annals of the University of Craiova, Electrical Engineering series, No. 41, ISSN 1842-4805, pp.40-47, 2017.

[15] S. Engell, G. Frehse, E. Schneider (Coordinators). Modelling, Analysis, and Design of Hybrid Systems, Lecture Notes in Control and Information Sciences 279, Springer-Verlag, ISBN: 3-540-43812-2, pp.3-36, 2002.

[16] Hamdi, F., Manamanni, N., Messai, N., Benmahammed, K., - Hybrid Observer Design for Switched Linear Systems Using Differential Petri Net, Proceedingsof the 16th Mediterranean Conference on Control and Automation Congress Centre, 978-1-4244-2505-1 ©2008 IEEE, pp. 95-100, Ajaccio, France, June 25 – 27, 2008.

[17] Emilia Villani, P. E. Miyagi, R. Valette. Modelling and Analysis of Hybrid Supervisory Systems; a Petri Net Approach, Springer-Verlag, ISBN 978-1-84628-650-6, pp. 1-13, 2007.

[18] J. Zaytoon (Coordinator). Systèmes dynamiques hybrides, Hermes-Science,2001.

[19] J. Wang, "Petri nets for dynamic-event driven system modeling", In: Fishwick, P.A. (ed.) Handbook of Dynamic System Modeling, Chapman & Hall/CRC, Taylor & Francis Group, Boca Raton, chap. 24, pp. 24-1–24-16, 2007.