Int J Adv Manuf Technol (2008) 36:1180–1190
DOI 10.1007/s00170-006-0924-5

ORIGINAL ARTICLE

Modeling extended Petri nets compatible with GHENeSys IEC61131 for industrial automation

José Reinaldo Silva · Israel Benítez · Luisa Villafruela · Oriol Gomis · Antoni Sudrià

Received: 3 May 2006 / Accepted: 28 December 2006 / Published online: 30 January 2007

Abstract Petri net (PN) modeling is one of the most used formal methods in the automation applications field, together with programmable logic controllers (PLCs). Therefore, the creation of a modeling methodology for PNs compatible with the IEC61131 standard is a necessity of automation specialists. Different works dealing with this subject have been carried out; they are presented in the first part of this paper [Frey (2000a, 2000b); Peng and Zhou (IEEE Trans Syst Man Cybern, Part C Appl Rev 34 (4):523–531, 2004); Uzam and Jones (Int J Adv Manuf Technol 14(10):716–728, 1998)], but they do not present a completely compatible methodology with this standard. At the same time, they do not maintain the simplicity required for such applications, nor the use of all-graphical and all-mathematical ordinary Petri net (OPN) tools to facilitate model verification and validation. The proposal presented here completes these requirements. Educational applications at the USP and UEA (Brazil) and the UO (Cuba), as well as industrial applications in Brazil and Cuba, have already been carried out with good results.

Keywords Petri nets · Modeling · Design systems · Programmable logic controllers · Verification · Validation · Sequential control

J. R. Silva
Mechatronics Department, SP—University of Sao Paulo,
Sao Paulo, Brazil
e-mail: reinaldo@usp.br

I. Benítez · L. Villafruela
Electrical Engineering Faculty, University of Oriente,
Santiago de Cuba, Cuba

I. Benítez
e-mail: ibenitez@fie.uo.edu.cu

L. Villafruela
e-mail: lvilla@fie.uo.edu.cu

O. Gomis (✉) · A. Sudrià
CITCEA-UPC Technical University of Catalonia,
Barcelona, Spain
e-mail: gomis@citcea.upc.edu

A. Sudrià
e-mail: sudria@citcea.upc.edu

1 Introduction

Modern industrial automation integrates all of the control systems with the enterprise automation systems to support the requirements of electronic commerce. Therefore, all modern industrial automation design increases in complexity. For that reason, automation projects require formal methods for automation design. For example, for many automated manufacturing processes in large-scale and long-distance distributed systems, remote monitoring is crucial to achieve guaranteed normal operations [1].

The scientific community [2–5] is currently occupied with searching for formal automation design methods which have to allow the direct conversion of formal methodology in programming applications in order to optimize the design cycle [3, 4]. This is a more professional and scientific way to achieve effective and safe applications from initial stages of design, avoiding the unnecessary expenses of time and resources in automation projects.

The formal study about discrete event controller design has four fundamental techniques: Automata, Petri nets (PNs), queuing networks, minimax and other algebraic methods [6]. Among these, we are interested in the PN methodology [7–9], as it has more advantages in graphical and mathematical simplicity compared to the other methods.

At the present time, a great number of works have appeared dealing with formal methods in the verification and validation of automation designs in processes of flexible manufacturing, batch processes, discrete event systems (DES) supervisory, etc. Many of these systems use programmable logic controllers (PLCs) [10] in their implementation. For that reason, several of these lines include PNs in automation model development with PLCs [2–5, 11–17]. However, the analysis and design methods differ according to each model's particularities and research lines.

Holloway et al. [15] express that the main current lines of controller design for discrete events systems on PNs are:

1. Controlled behavior approach: where the model includes the behavior of the plant and controller combined. Therefore, when the desired controlled behavior is obtained, it is necessary to extract the controller logic for implementation [18].
2. Logic controller approach: this consists of the direct design and implementation of a controller for the plant based on defining of the I/O behavior required to achieve the desired controlled behavior of the system. Here, it is necessary to give input signals to the controller, so that it executes the required actions; therefore, it is necessary to validate it for simulation [11].
3. Control theoretic approach: this approach uses Ramadge and Wonham's paradigm of classical controller DES design [19], but merges it with PN modeling. Given a model of the plant dynamics and a specification for the desired closed-loop behavior, the objective is to synthesize a controller to achieve the specifications [2, 15].

In [20], several PN techniques for DES supervisors are discussed. Here, two important groups of supervisors based on PNs are extracted:

– Mapping supervisor: the control politics is efficiently computed by an online controller as a feedback function of the system marking
– Compiled supervisor: the control politics is represented as a net structure

The second Giua supervisor along with the third Holloway approach is the best methodology because it is quicker when there is no need for online controller computation. Both systems (non-controlled and controlled) use PNs and allow to use modular models similar to IEC61131 standard blocks. The use of a control theoretic approach permits to apply forbidden states and desired string theory in the automation design process.

In Uzam's doctoral theme, a controlled system specification is used [21]. It is very well expressed in Automata theory and is related to two specification categories that appear in the supervisor control literature:

– Forbidden state problem: here, the control specifications are expressed as forbidden conditions that must be avoided [21]
– Forbidden (desired) string problem: here, the control specifications are expressed as a sequence of activities that must be provided, while not allowing the undesired sequence of activities to occur [21]

The supervisor's synthesis is carried out using these two specifications categories; that is to say, all events which do not provoke forbidden situations are allowed to happen. In PNs, the same model can be used for functional properties analysis, the behavior evaluation, as well as for the systematic construction of discrete event controllers [22].

Several focuses to address the forbidden states and desired chains problems on PN exist, beginning with one that contains the problems in a single class, called the generalized mutual exclusion constraints (GMEC) [23] method, until which, works by the use of a general extended class of controlled Petri nets (CtrlPN) [15]. But, these methodologies do not permit to create practical applications efficiently. It is part of the gap that exists between the theories of discrete event control systems (DECS) and their industrial implementation [5]. For that reason, a new methodology to permit the closing, and, possibly, the disappearance, of this gap is required, but it should include the good results of other currently existing methods.

According to Holloway et al. [15], the plant controller's synthesis, modeled by PNs that use a control theoretic approach, have two main lines:

– State feedback control: based on the addition of control places to the PN, creating the CtlPN (controlled PN)
– Event feedback control: based on the assignment of events to transitions, creating the LabPN (labeled PN)

CtlPN is a 5-tuple Nc=(N, C, B, M0, MF), where N=(P, T, F) is an ordinary Petri net (OPN), C is a finite set of control places, disjoint from P, T, while $B \subseteq (C \times T)$ is a set of directed arcs connecting control places to transitions, M0 is the initial marking, and MF is a finite group of the final marking.

LabPN is a 5-tuple G=(N, Σ, l, M0, MF), where N=(P, T, F) continues being an OPN, Σ is a finite set (alphabet) of events, l: $\Sigma \rightarrow T$ is a function that assigns events to transitions, and M0 and MF continue indicating the initial and final markings, respectively.

In Sect. 2, three very often used variants of these two lines are compared in the modeling of PLC automation applications. It has the purpose of determining the best variant that comes closer to the international standards of

PLC programming, so that it allows the necessary use of popular formal automation design cycles. In Sect. 3, a complete methodology using this variant is presented. It allows full industrial applicability. A brief explanation of some applications of this methodology is introduced in Sect. 4. Finally, our conclusions are summarized in Sect. 5.

2 Petri net models

In the LabPN line, a *signal interpreted PN* model exists [3, 4]. This model is defined by SIPN=(P, T, F, M0, I, O, φ, ω, Ω), where (P, T, F, M0) is an OPN with initial marking M0, I is a logical input set, O is the logical output set, φ associates all transitions to a Boolean condition of I, ω associates all places to the values (0, 1, –) of an output O, and Ω is a function that combines the values obtained for each output with places to avoid conflicts and undefined conditions (Fig. 1c).

In SIPN, the Ω function associates each output to places of values, according to the followed conditions: undefined (–), zero (0), one (1), contradictory (c), redundant to zero (r0), redundant at 1 (r1), and combinations of c (c0, c1, c01) [3, 4]. This definition increases the complexity of the output PN association. Here, a fourth transition (firing) rule also appears as an iteration (practically instantaneous) of the firing sequences until a stable state appears (neither transition is fire). Everything is included inside a PLC cycle. This is a complicated net behavior moving away from the OPN.

In a central position between the CtrlPN and LabPN lines, the *Automata PN* (APN) [2] has been developed with the intermediate places method. It is defined as APN=(P, T, Pre, Post, In, En, X, Q, M0), where P are places, T are transitions, Pre: (P×T)→N and Post: (T×P)→N are ordinary arcs between P and T with associated weights, but In: (P×T)→N are inhibitor arcs and En: (P×T)→N are enable arcs that represent the connection of auxiliary (intermediate) places associated with the presence or absence of control signals on the system, X is the set of conditions associated to the transitions, Q is a set of actions assigned to the places (they can be impulse or level actions), and M0 is the initial marking (Fig. 1a). The sensor signals will be associated to the event conditions of transitions or to intermediate places with In or En arcs.

Inside the CtrlPN line is the *extended PN GHENeSys* [24]. It can be defined as a 6-tuple N=(L, A, F, K, M, Π), where the elements of L are called places and are compounded by the union of the B and P sets (boxes and pseudoboxes, respectively); B is the set of normal places, but P are places of permanent marking {0, 1} only modified by external events, and the elements of P are linked with the controlled transitions by means of inhibitor or enable arcs. The elements of set A are transitions (activities). F is the flow relationship (F⊆(L×A)∪(A×L)) and its elements are called arcs of unitary weight. K: B→N^+ is the function capacity that indicates the maximum capacity allowed in each box. M: L→N^+ is the initial marking of the net respecting the capacities of each place. Π is a function that differentiates simple elements (L and A, assigning them a value of 0) of the macro-elements (elements that represent subnets; a value of 1 is assigned) (Fig. 1b).

To this initial definition [24], a Q function is added to associate level actions (assignment) to some boxes (B) or impulse actions (set or reset) to the firing of some activities (A).

Other variants are presented in [5, 17, 18, 25]. There have been no significant differences compared to the previously cited lines and, therefore, they are not considered in this work. For example, Lee et al. [25] use an OPN very close to the APN, but of single market nature, limiting it to simple sequential applications, which includes an

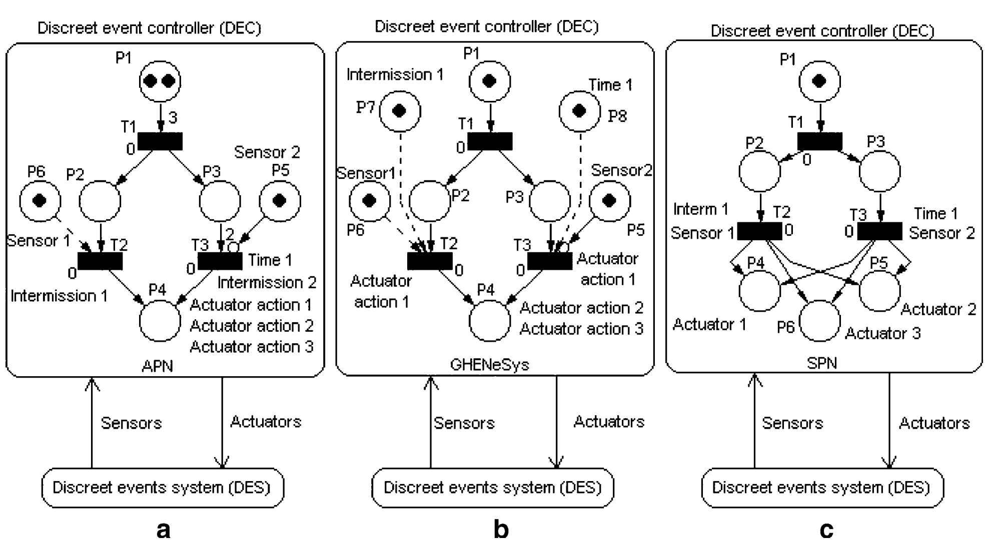

Fig. 1 Discrete events system (DES) controller modeled using: **a** Automata Petri net (APN); **b** GHENeSys; **c** single interpreted PN (SIPN)

operation timer in process sequence transitions that is far more functional than IEC61131 timer blocks. Lee and Hsu [17] introduce the IDEF0 modeling stage before the simplified PN controller (SPNC) model (single marking also); at the end of this stage, they use the token passing logic (TPL) translation, similar to APN and GHENeSys translation.

The three models studied here (APN, GHENeSys, and SIPN) use the particularity of assigning actions toward processes to places. It allows the extension of a model to control the process performance, and it is an important advantage compared to the CtlPN and LabPN definitions of Holloway et al. [15]. However, in the SIPN case, it associates the output and not the output action; therefore, an additional function (Ω) is necessary, and it also increases the state explosion.

These three lines also differ in the form of signal treating of the process state's sensors. The SIPN variant (near to LabPN) limits the OPN's graphical representativeness, as it does not represent the sensors' actions in their structure neither graphically nor mathematically; it only appears as transition events (e.g., Sensor1 and Interm1 are events associated to T2 in Fig. 1c). The APN represents sensor actions graphically if the use of auxiliary places (P6 associated to Sensor1 in Fig. 1a) is considered, but only graphical analysis design methods are used. However, GHENeSys allows a graphical and mathematical representation of such sensors' signals. Graphical sensor representation is the use of auxiliary places for all control signals (e.g., P5–P8 in Fig. 1b). The mathematical method uses an additional diagonal matrix D in the state equation [24]. This allows its inclusion in the structural analysis (e.g, mathematical calculation of S and T invariants). A graphical representation of sensors' actions permits a real simulation of it. This provides better facilities to study the GHENeSys PN models for DES supervisor (controller) design.

For all of the above mentioned, GHENeSys PN models are considered to be more explicit and, as a CtlPN variant, it is near to the classic PN theory, but extending this theory with the addition of inhibitor and enable arc facilities to link auxiliary places to transitions (in APN and GHENeSys PNs). The sensor's action is guaranteed on the PN model with marking persistence in those special places (they are not affected by transitions firing). Other CtlPN models (e.g., Lee and Hsu [17]) use auxiliary places (sensor state in 17), but only represent sensor signals to control transitions in 1-bounded PNs. GHENeSys PNs are bounded to the OPN field (it is considered to be in the range between C/E and P/T) to permit the application of many mathematical analysis tools of these PNs, and, therefore, they have an advantage compared to APNs. APNs cover a wide field (they can use arcs' weight and different tokens); it allows the use of colored nets in APN models, but they lose the power of the simple mathematical calculation of an OPN. Therefore, APN analysis methods are only based on reachability graph (RG) analysis and are directly influenced by the state explosion. GHENeSys nets allow the use of analysis tools, RG, and state equations.

It could be considered that the GHENeSys' restriction to OPNs limits its applicability to complex systems, but this is not true since, in these systems, an object-oriented variant of GHENeSys [24] can be used. This variant maintains the OPN facilities by using a combination of ideas from the object-orientated paradigm with OPN [24], without increasing the net's complexity with compensated arcs or colored nets.

GHENeSys use includes a great field of PLC applications and guarantees many analysis tools. Also, GHENeSys nets are near to achieving simplex models because pseudoboxes (auxiliary places) not only represent process sensors' measurements, but they also represent information states coming from other subnets or parts of the same net [24]. Therefore, GHENeSys PNs allow more effectiveness using the classic tools of OPN analysis and design.

The GHENeSys variant presented in this work also differs from APNs in the association of impulse and level actions (Fig. 2). In GHENeSys, the impulse actions are associated to activities (transitions) firing and level actions are associated to boxes (places with variable markings). It permits not requiring to separate action type in the Q function and it comes closer to the OPN's dynamic nature. Impulsive actions are closer to transitions firing (it happens instantly and then disappears), while level actions stay the whole time required by the system, the same as place activation. In APNs, all signals are only associated to places; therefore, different action types are required.

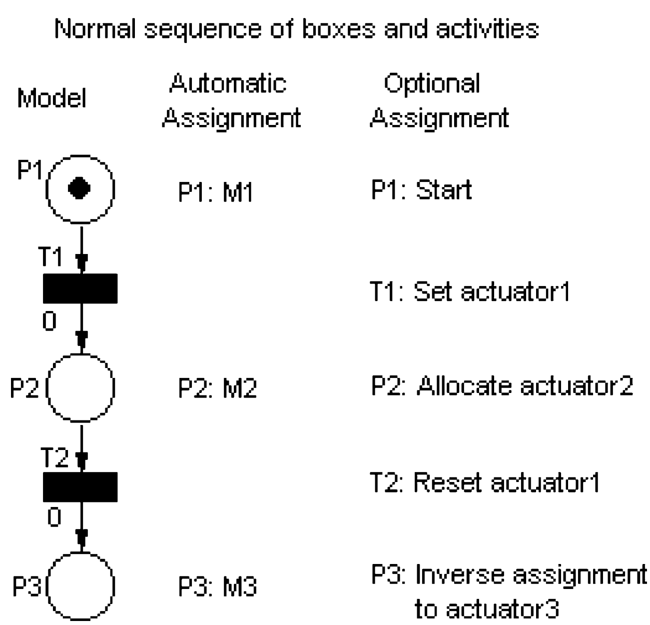

Fig. 2 Binary sequence modeling using GHENeSys

Also, subnet definitions (macros) in GHENeSys helps to model modularity and, therefore, permits their reusability and hierarchical nets conformation, reducing the state explosion effect. These macro-elements also allows GHENeSys to embrace the field of non-binary applications on PLCs without having to pass to high-level nets (like in APNs) and with a larger similarity to PLC languages that treats this aspect with modularity inside function blocks (Fig. 3).

A GHENeSys net allows the creation of standard module libraries that can build a bigger model based on typical control structures [26] and then to add that to a hierarchy. It permits model design and their translation to PLC languages.

3 Methodology proposal

For the use of extended GHENeSys nets in the process of automation systems design with PLCs, a general methodology is proposed. It includes the following steps:

1. Analyze the process system and the functional requirements to control it.
2. Determine more quantity of control system functional units, without coinciding with the equipment of their future implementation. This permits to create different subsystem elements that form parts of the PN plant model.
3. Define internal actions and interdependencies of each functional unit and the place in the hierarchical level according to these interdependencies and their function in the control system. This permits to relate all subsystems.

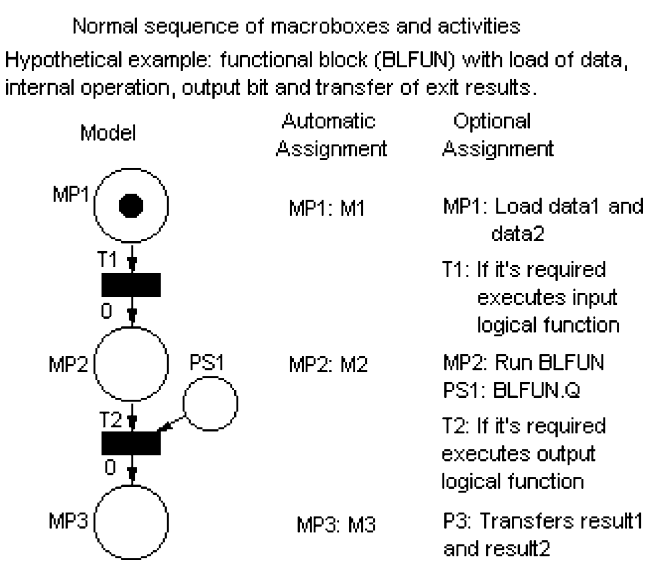

Fig. 3 Modeling an analogical sequence using GHENeSys

4. Use top-down design to establish a hierarchical model based on macro-elements for each functional unit (subsystem), establishing a hierarchical GHENeSys net for the whole model of the control system. The relations between subsystems (macro-elements) create a new subnet at a high hierarchical level.
5. Realize bottom-up refinement of the model to detail the structure of each subnet (macro-element), looking for their internal simplest representation in an OPN type, if it is possible. If more complex nets are required, new subnets should be created (macro-elements) for only this situation. It allows applying simple OPN models in all system-controlled design methods and only complex tools in small subnets. The typical structural DES models are used to create internal subnet configurations.
6. Use the C-TPM method [2] adapted to GHENeSys nets for each subnet control design, applying the solution of the desired sequence and prohibited state problems. It utilizes auxiliary places (pseudoboxes) with inhibitor and enable arcs to prepare preconditions of the controllable transition.
7. Classify each subnet and determine their functional (liveness and boundedness) and structural properties (S- and T-invariants). Apply a preferably simple reduction rules method to revise the net structure. This step develops in the structural PN (without pseudoboxes).
8. Improve the model, eliminating deficiencies detected and reintroducing pseudoboxes.
9. Reclassify each modified subnet and predetermine their properties, carrying out adaptations that allow their required execution.
10. Simulate the work of each subnet and verify the execution of user's functional requirements.
11. Improve and repeat steps 7 to 10 if required.
12. Check peculiar characteristics of equipment for implementation to model the subnet that carries out the synchronization of functional communications.
13. Translate model to an IEC61131-compatible program [27], preferably SFC, in all of its levels, and considering the user's approach to select another type of language (ST, IL, LD) for the basic subnets (macro-elements).

In this formal design, user requirements formalization is conformed by steps 1 to 6. Steps 7 to 9 contain verification of the functional and structural properties of the model. In steps 10 to 12, model validation is carried out according to the desired behavior for controlled systems and implementation particularities. All of this guarantees that the result is an efficient and safe model.

The last step is the controlled system implementation by translating the previously verified and validated model to PLC programming languages that are IEC61131-compatible. Translation peculiarities will be addressed in further work, as they are not in the scope of the present paper.

3.1 Design methodology of GHENeSys nets

Initially, the process and the controlled functional requirements analysis are carried out. It allows to separate the whole system in the largest quantity of possible functional units. These constitute modules at different hierarchical levels which have the largest independence as possible, and whose interactions will be presented in the higher hierarchical module that contains them. This proposal is in correspondence with modeling methods proposed by several authors [13, 28, 29], but not developed in the way leading to IEC61131 compatibility, such as that which occurs in the GHENeSys model.

Functional units contain operations and resources that are dedicated to system-specific functions, creating subsystems. Therefore, it is necessary to define their internal relationships, to create an internal structure of the module. This internal structure is created and defined by different stages of the subsystem process, which are places, and the events that develop the state changes, which are transitions. These places and transitions are related by utilizing typical DES structured models (sequential, conditional, parallel, and others). Each subsystem forms reusable libraries of different processes, such as flexible manufacturing systems automation design [30, 31] require a flexible manufacturing library (robot, machine, conveyor, buffer, etc). Besides, interrelations with other modules are defined that will be modeled at a higher hierarchical level. Then, each subnet goes by other design steps and modification.

The hierarchical GHENeSys design proposed here also agrees with the modeling approach proposed by Luca Ferrarini [13], where bottom-up and top-down techniques in PN modeling are incorporated.

A non-controlled process model can be created using standard PN module combinations that represent the different model functions. This is known as *modulate modeling*. These structures (sequential, conditional, iteration, parallel, resource share, synchronization, producer–consumer relation, buffer) appear frequently [2, 24, 32].

The process' and user's functional requirements can be formalized to have a hierarchical non-controlled process model in the PN. It is the same as the six design methods presented in [2, 33] (arc inhibitor, arc available, intermediate place, APN-SM, OR-TPM, C-TPM methods). But, in our methodology, these methods are restricted to the extended OPN to permit the utilization of all of its tools. It utilizes a typical control structure library, such as different selection structures.

To created well structured resulting programs, it is necessary to use proper structures (only one input and one output, which must include an input/output route) [34]. It is the same as well formed PN definitions [35]. Therefore, in PN models and programs, the liveness (avoiding deadlocks) and operation without hold is guaranteed [35].

The C-TPM method [2] (adapted for use on GHENeSys nets in this work) has control politics based on making the firing of the controllable transitions depend on the non-controlled model. This way, the transitions firing is only enabled when allowing the movement of the system (it is only in the space of maximally permissible states). This control guarantees that the plant does not fall into the forbidden states of the system.

Each model transition can be associated to the occurrence of external events in two ways: controllable (can be disabled through the controller) or uncontrollable (cannot be disabled through the controller). However, OPNs do not allow sensors and actuators association; for that reason, PN extensions are utilized. For example, in GHENeSys pseudoboxes with enable or inhibitor arcs, controllable transitions are used and actuators are modeled by associating control actions to places (boxes) and/or transitions (activities). It constitutes a new form of association. The relations are created using the desired sequence, forbidden state problems, and other control requirements.

Introducing pseudoboxes and actions association, the control politics required to culminate the model design of a controlled system is completed. Next, the verification and validation stages to obtain the final model for implementation are executed.

3.2 Verification using GHENeSys

Verification using SIPNs [36] applies PN analysis and verification methods based on control algorithms. The interpreted PNs (IPNs) are replaced by PN elements (without firing conditions and actions association) to have an OPN. After substitution, OPN properties are determined by the reachability graph method. Finally, properties achieved in OPN assure an efficient and boundedness model after associations reestablishing.

The methodology presented here proposes a similar variant, but applied to the GHENeSys case, which the output redundancy and contradiction problem of SIPNs are not present because it works by assigning actions and not directly outputting to places. The GHENeSys case uses the same OPN model transformation, eliminating pseudoboxes and their interconnection arcs, to carry out property verification, such as liveness and boundedness on an OPN. In this way, source places achieving a pure PN are eliminated—they fulfill the necessary condition of liveness outlined by Murata [9] (to not have sources and drains).

Most of the automation applications with PLCs can be considered inside of free-choice PNs (FC-nets). Therefore, liveness and boundedness analysis of controlled models proposed on GHENeSys can use the methods proposed in [35]. According to Chap. 7 (*Reduction and synthesis*) of [35], the reduction rules give an alternative method of well formed analysis in FC-nets. The well formed checking algorithm can be transformed easily in a liveness and boundedness checking algorithm of free-choice systems using Theorem 6.17 [35], which allows to apply reduction rules to check for well formation.

In [13] also, the use of simple reduction rules is proposed as effective tools to allow property model analysis in complex PN systems.

These aspects are also treated in [14, 37], but adding other rules to reduce the complex models in the PN. All of them maintain liveness and boundedness properties between the initial PN and the reduced resultant PN.

If a subnet is more complex, then the range theorem [35] variant can be applied. This theorem gives a complete verification method that allows to solve the polynomial definition problem of liveness and boundedness in the FC-net through well formedness and FC-net characterization.

3.3 Validation using GHENeSys

In [13], the analysis methods of the models in PNs are explained, including the confirmation of functional correspondence existing among the model with the specifications of original requirements (typically expressed informally), which is proven in the validation. To achieve this correspondence requires experience in the modeling and knowledge of the techniques that help the model's construction. Therefore, the completeness is also included in the specification of requirements. These last correspondences are generally expressed like relationships of I/O of the system. There are inputs that are not defined in the initial requirements and they should be completed. Another important aspect is the consistency of the requirement specifications. Consistency does not exist (inconsistency) when a combination of inputs gives several combinations of outputs. All of this should be proven during the model's validation.

In [13], the simulation of discrete events is considered as another way for the checking up of the system properties. It is used as an execution algorithm to run the net. Among the disadvantages, one can find it to be an extensive technique, having a great consumption of time, and that it shows the presence of undesirable properties, but it does not prove the model to be correct in general cases. However, it remains as a proposal for many authors. Specifically, in [14], they consider it as a tool of behavioral analysis when the limitations of computers do not allow the generation of the accessibility graphics. Desrochers and Al-Jaar [14] yield that the simulation allows to observe how many times a place is marked or not and to calculate this probability. Therefore, the simulation of token flow can generate important statistical analysis of the PN behavior. Much more interesting is when the simulation tool allows model execution in real time.

The GHENeSys validation of this methodology deals with the simulated study of the behavior of the controlled models, verifying the execution of restrictions of forbidden states and the desired controller's sequence.

Considering the hierarchical modularity on GHENeSys design proposed in this work, the analyzed subnets should be separate nets (one input, one output, and I/O markings that cover all nodes), and also, during the verification, the subnet liveness should have been achieved. Therefore, as these subnets do not have large dimensions, the pseudo-boxes can be reinstalled to carry out the simulation of their behavior when evaluating user's specifications execution for the controlled system.

Of the whole analysis of this proposed methodology, we can conclude that, as a result, we have a model as a PN, created using proven methods, and the verification and proposed validation allow the guaranteeing of the boundedness and efficiency of the PN model in front of user's requirements for a specific application.

4 Educational and professional applications

At the University of Oriente, Santiago de Cuba, Cuba, an educational system has been developed to allow the use of this methodology in the facilities for several laboratory practices (gassy panel pressure control, liquid level control, flow control) in the Electrical Engineering Faculty [38]. At the Amazon State University, UEA, Manaus, Brazil, a laboratory flexible manufacturing cell uses GHENeSys methodology in the graduate and postgraduate courses of Mechatronic Engineering [39]. This allows the use of formal methods on OPNs for automation system design in different student groups.

In automation projects, there are always certain necessary classic structures, such as a selector of two or more positions, a control of two positions (ON–OFF), comparisons to verify alarms, or to control several positions with hysteresis, etc. These typical structures can be modeled with this methodology to have typical reutilizable subnet libraries in the new projects.

In the PN models of pressure control in a laboratory tank of the UO (gassy panel), these subnets were created to be included in a typical structures library which is used in the PN model design of other practical laboratories or for other more complex processes. This library is employed in the practical classes of the subjects of Automation Engineering education. The subnets of the library can be incorporated by the students in their own projects, such as the role of

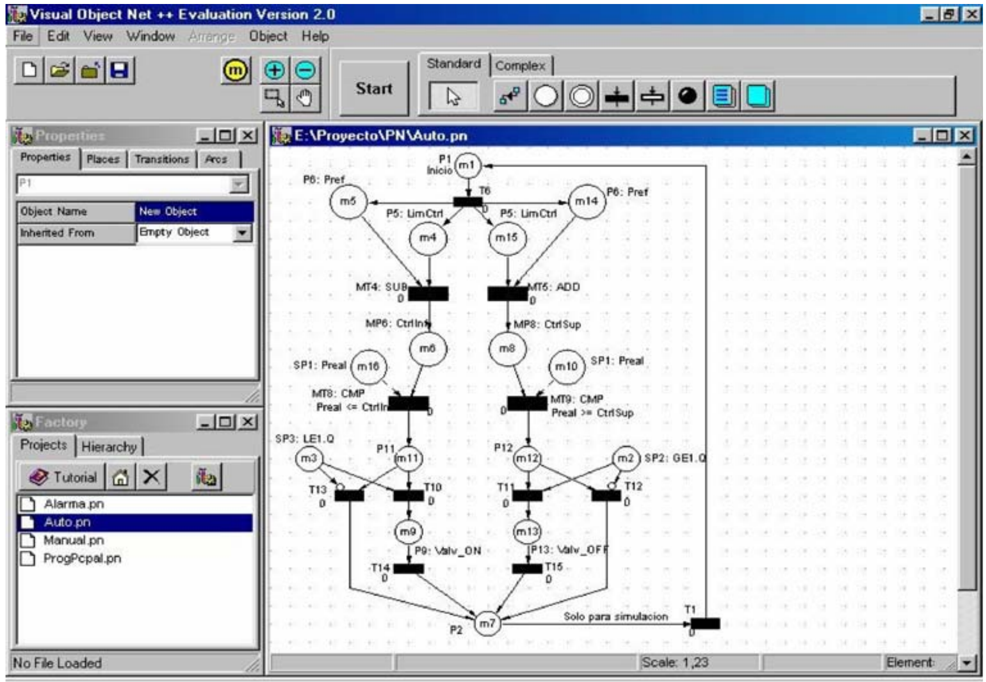

Fig. 4 Control of two positions with hysteresis at the UO laboratory, Santiago de Cuba, Cuba, using Visual Object Net (VON)

learning and familiarization with these methods. To create this library of models in OPNs, GHENeSys was used with Visual Object Net (VON) [40].

Figure 4 shows the PN module used in the two-position pressure control of a compressed air tank (part of the gassy panel installation at the UO laboratory).

The two branches of the model uses macro-elements (macro-places and macro-transitions) to carry out the addition and subtraction of permissible limits, compared with the real value of gassy pressure. At the end, the right decision was made for the gaseous control.

At the UEA, the laboratory installation has a pneumatic manipulator, conveyor, robot, and machine tool working together. All control programs within the PLCs and the robot controllers, are modeled and programmed with the GHENeSys methodology. For example, the activities synchronization between the robot and tool machine (loading and unloading) can be modeled by the GHENeSys methodology, with initializing by the machine tool and robot subsystem models (Fig. 5a,b). The plant models are created by defining their interrelations (Fig. 5c).

Developing top-down design and bottom-up refinement are defined some macro-elements to develop low hierarchy subnets. Using the C-TPM method adapted to GHENeSys nets and applying the solution of the desired sequence and prohibited state problems, the auxiliary place (pseudo-boxes) with enable and inhibitor arcs are added (Fig. 6a). This model permits properties verification and functional validation by adding timer simulations (Fig. 6b).

Starting from this educational experience, this methodology was used in collaboration with Cuban automation companies to improve the automation projects of sugar shipment terminals of the Carúpano, Guayabal, and Mariel Ports. This has allowed savings for fast delivery and efficiency in the weighing of sugar in these Cuban ports. This methodology is now used in the industrial applications of Manaus (Amazonas, Brazil) by means of collaboration research projects between the University of Amazon State (UEA), USP (São Paulo), and UO (Cuba). Important results

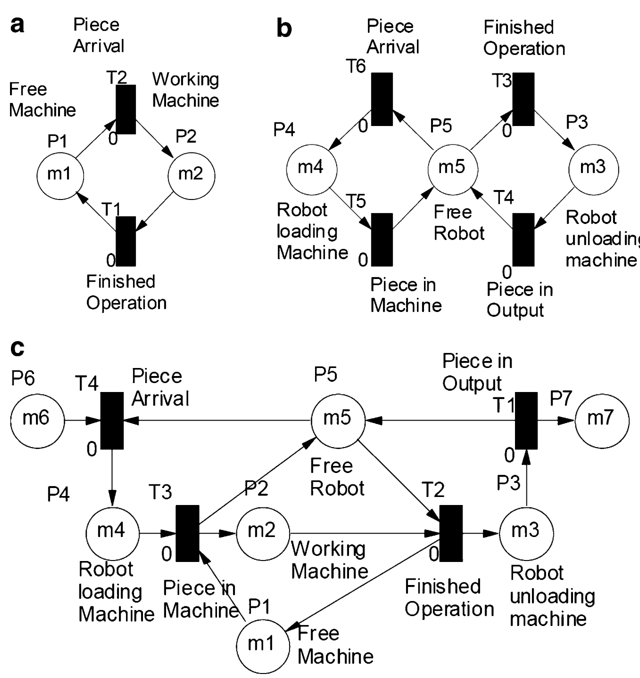

Fig. 5 First steps of the GHENeSys methodology at the UEA laboratory (Manaus, Brazil)

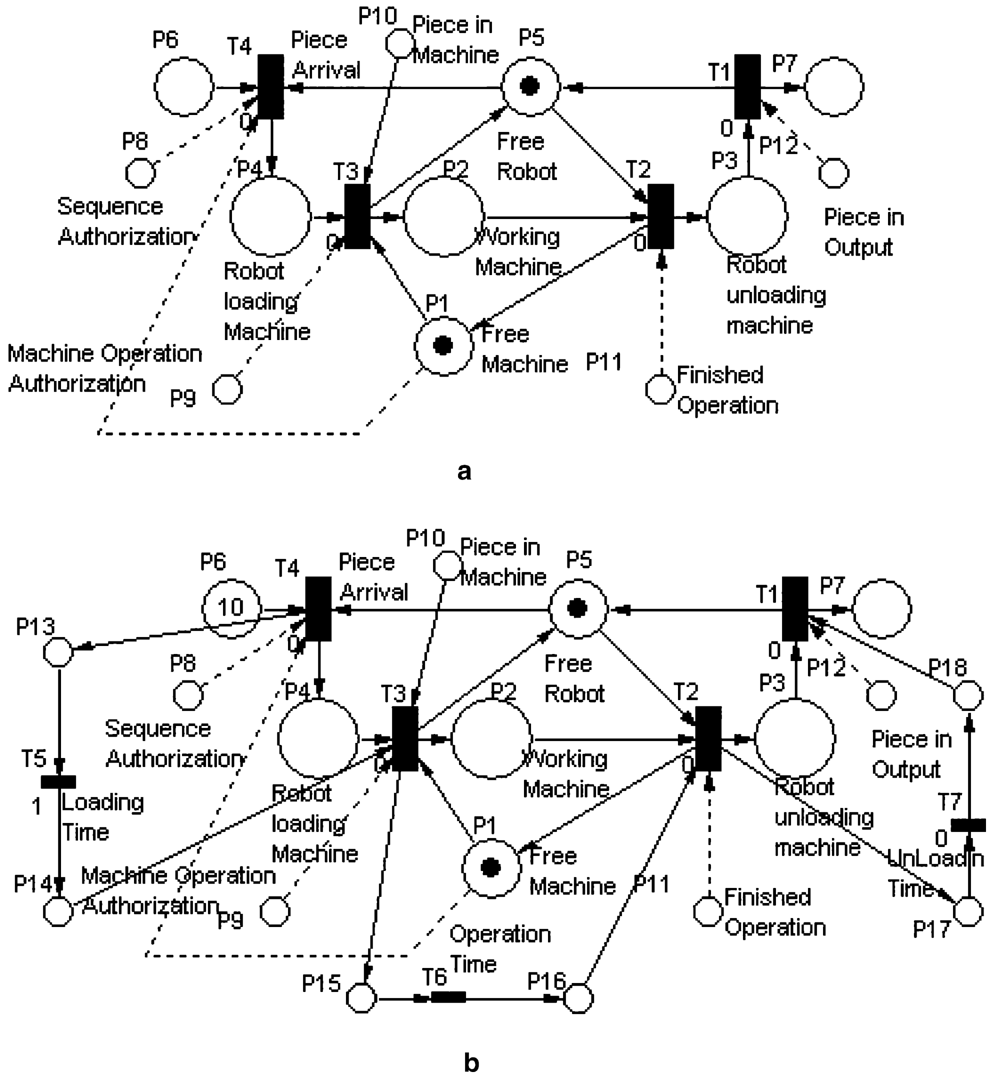

Fig. 6 Final steps in the GHENeSys modeling methodology at the UEA laboratory

have been obtained in two industries of the Manaus industrial pole.

One example of the use of GHENeSys to solve industrial automation problems was developed in the welding area of the automotive industry in the Manaus industrial pole. It has three welding robots (three areas with one circle inside, shown in Fig. 7a). The project began with the process and control requirement study. Figure 7a represents the layout of the welding cell in its initial stage, highlighting the operational movements. Therefore, it is very important to reduce the operation time in these movements. Figure 7b introduces a manipulator robot (circle with two rectangles which represents the robot's movements). The manipulator robot was introduced as a result of the GHENeSys model analysis.

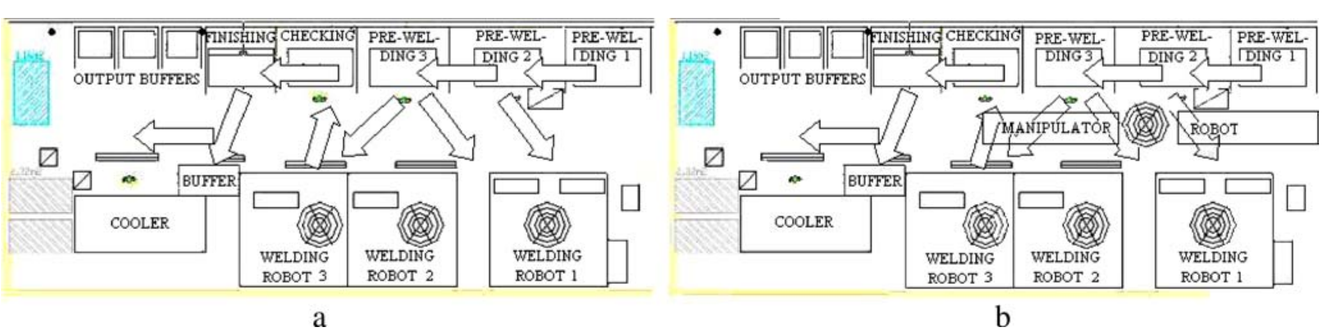

Fig. 7 **a** Initial and **b** final layout of a welding cell in the Manaus industrial pole

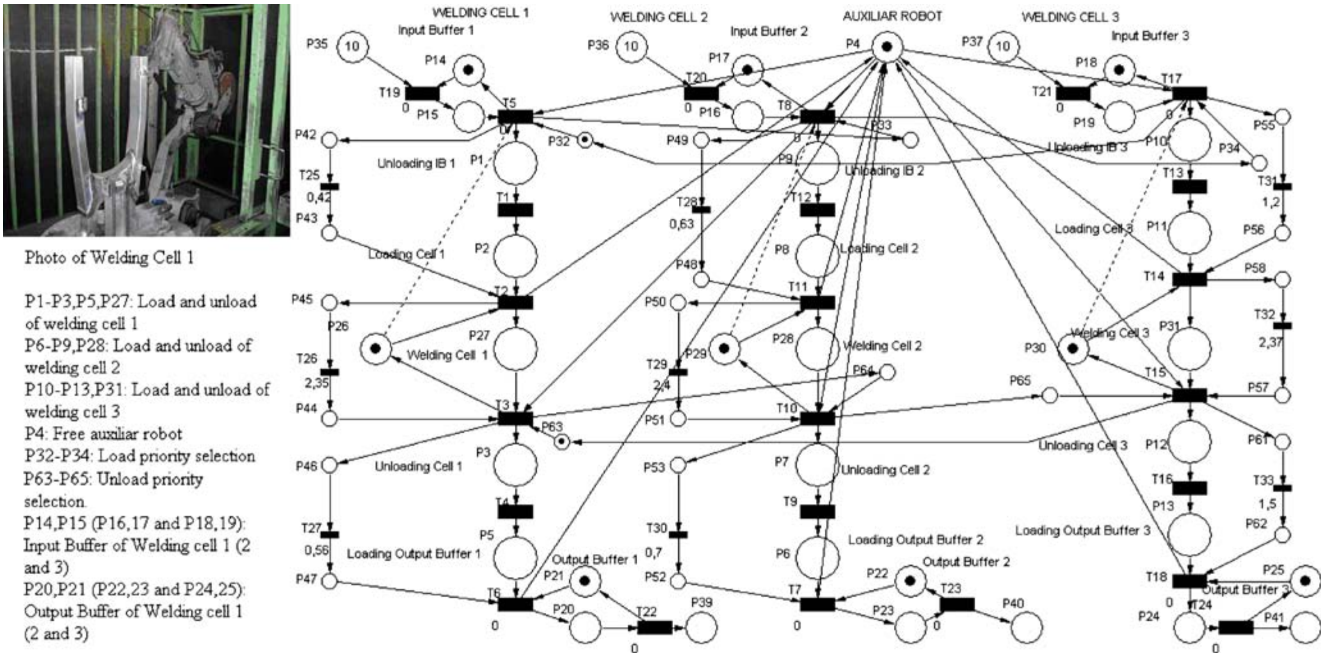

Fig. 8 Photo of the GHENeSys PN model of a manufacturing welding cell in Manaus

Figure 8a shows a photo of a robot with a piece placed on the revolving table. Figure 8b shows the GHENeSys PN model for the establishment of the operation sequence of loading and unloading the three welding cells with an auxiliary manipulator robot. The first column on the left represents the loading, working, and unloading of welding cell 1. It was created by searching the interactions of typical models of the operation machine and the manipulator robot using desired the operation sequence and forbidden situations. Columns 2 and 3 for the other cells were created in the same form.

A place P4 represents a free auxiliary manipulator robot. The GHENeSys PN model allowed operation synchronization and priority definition using real operation times obtained from experimental measurements. The use of auxiliary places, macro-elements, and enabling and inhibitor arcs of the GHENeSys methodology guaranteed modeling facilities and IEC61131 compatibility applied in a solution of an industrial problem. The GHENeSys verification and validation were very simple (reduction rules and simulation) and guaranteed by GHENeSys modularity facilities. Other automation PN models developing the same design stages are more complex and less IEC61131 compatible, encountering more difficulties and extending the project time.

5 Conclusions

A GHENeSys Petri net (PN) model is presented to develop programmable logic controller (PLC) applications. It uses the facilities of pseudoboxes, special arcs, and macro-elements of the GHENeSys PN, but adding a function of process actions association with places (boxes) and transitions (activities) as levels of impulse actions, respectively, in the resulting PN model. This allows the creation of a hierarchical model with high-level modularity, similar to IEC61131 PLC programs.

The C-TPM method is adapted for automation design on the GHENeSys PN model. It takes advantage of the graphical and analytical facilities of GHENeSys PN models to propose graphic methods (reduction rules) and analytic methods (state equation and range theorem) in model properties verification. It includes the use of subnet behavior simulation to validate user requirements for the controlled system. This constitutes a comprehensive methodology of design, verification, and validation of PN models for PLC automation that gives us tools to achieve the needed requirements in a wide range applications. It reduces the gap between discrete event control systems (DECS) theory and their industrial implementation.

The presented methodology is already used in laboratory practices in the UO (Cuba), USP (São Paulo, Brazil), UPC (Barcelona, Spain), and the UEA (Manaus, Brazil). But this methodology was created to be utilized to develop industrial applications because it allows the entire integration of design and implementation of automation systems with the simplicity that does not require an industrial specialist. Different industrial applications developed in Cuba and Brazil show these advantages.

Currently, our international research group is working to create all of the software support to convert this methodology into a powerful industrial automation tool.

References

1. Lee J-S, Hsu P-L (2004) Design and implementation of the SNMP agents for remote monitoring and control via UML and Petri nets. IEEE Trans Control Syst Technol 12(2):293–302
2. Uzam M, Jones AH (1998) Discrete event control system design using automation Petri nets and their ladder diagram implementation. Int J Adv Manuf Technol 14(10):716–728
3. Frey G (2000a) Analysis of Petri net based control algorithms—basic properties. In: Proceedings of the American Control Conference (ACC 2000), Chicago, Illinois, June 2000. pp 3172–3176
4. Frey G (2000b) Automatic implementation of Petri Net based control algorithms on PLC. In: Proceedings of the American Control Conference (ACC 2000), Chicago, Illinois, June 2000, pp 2819–2823
5. Peng SS, Zhou MC (2004) Ladder diagram and Petri-net-based discrete-event control design methods. IEEE Trans Syst Man Cybern, Part C Appl Rev 34(4):523–531
6. Koussoulas NT (1994) Discrete event dynamic systems: which model to choose? In: Bryds MA, Malinowski K (eds) Computer aided control system design: methods, tools and related topics. World Scientific, Singapore
7. Reisig W (1982) Petri nets: an introduction. Springer, Berlin Heidelberg New York
8. Thiagarajan PS (1986) Elementary net systems. Lect Notes Comput Sci 224:27–59
9. Murata T (1989) Petri nets: properties, analysis and applications. Proc IEEE 77(4)541–580
10. McBride A (2001) PLC. Get more than what you see. Control Eng J 48(2):95–102
11. David R, Alla H (1992) Petri nets and Grafcet: tools for modelling discrete event systems. Prentice Hall, New York
12. DiCesare F (1993) Practice of Petri nets in manufacturing. Chapman & Hall, London, UK
13. Zhou M (1995) Petri nets in flexible and agile automation. Kluwer, Boston, Massachusetts
14. Desrochers AA, Al-Jaar RY (1995) Application of PN in manufacturing systems. IEEE Press, Piscataway, New Jersey
15. Holloway LE, Krogh BH, Giua A (1997) A survey of Petri net methods for controlled discrete event systems. Discrete Event Dyn Sys 7(2):151–190
16. Lucas MR, Tilbury DM (2002) Quantitative and qualitative comparisons of PLC programs for a small testbed with a focus on human issues. In: Proceedings of the American Control Conference, Anchorage, Alaska, May 2002
17. Lee J-S, Hsu P-L (2005) A systematic approach for the sequence controller design in manufacturing systems. Int J Adv Manuf Technol 25(7–8):754–760
18. Zhou MC, DiCesare F (1993) Petri net synthesis for discrete event control of manufacturing systems. Kluwer, Boston, Massachusetts
19. Ramadge PJG, Wonham WM (1987) Supervisory control of a class of discrete event processes. SIAM J Control Optim 25 (1):206–230
20. Giua A (1996) Petri net techniques for supervisory control in discrete event systems. In: Proceedings of the 1st International Workshop on Manufacturing and Petri Nets, Osaka, Japan, June 1996, pp 1–12
21. Ramadge PJG, Wonham WM (1989) The control of discrete event systems. Proc IEEE 77(1):81–97
22. Zurawaski R, Zhou MC (1994) Petri nets and industrial applications: a tutorial. IEEE Trans Ind Electron 41(6):567–583
23. Giua A, DiCesare F, Silva M (1993) Petri net supervisors for generalised mutual exclusions constraints. In: Proceedings of the 12th IFAC World Congress, Sydney, Australia, July 1993, vol I, pp 267–270
24. González P, Silva JR (2001) GHENeSys: uma rede estendida para a modelagem, analise e projeto de sistemas complexos. In: Proceedings of the Brazilian Symposium on Intelligent Automation (SBAI 2001), Sao Paulo, Brazil, November 2001
25. Lee G-B, Zandong H, Lee J-S (2004) Automatic generation of ladder diagram with control Petri net. J Intell Manuf 15(2):245–252
26. Hasegawa K, Takahashi K, Masuda R, Tange K, Suzuki T (1987) Simulation of discrete production systems based on mark flow graph. Syst Sci 13:105–121
27. Lewis RW (1998) Programming industrial control systems using IEC 61131-3. IEE Control Engineering Series 50. IEE, London, UK
28. Best E, DeCindio F, Hopkins R (1990) DEMON design methods based on nets. ESPRIT BRA (basic research action) #3148. EATCs Bull 41:487–505
29. Bernardinello L, de Cindio F (1992) A survey of basic net models and modular net classes. Lect Notes Comput Sci 609:304–351
30. Uzam M (2002) An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions. Int J Adv Manuf Technol 19(3):192–208
31. Park E, Tilbury DM, Khargonekar PP (2001) A modeling and analysis methodology for modular logic controllers of machining systems using Petri net formalism. IEEE Trans Syst Man Cybern, Part C Appl Rev 31(2):168–188
32. Hasegawa K (1996) Modeling, control and deadlock avoidance of flexible manufacturing systems. In: Proceedings of the 11th Brazilian Congress of Automation (XI Congresso Brasileiro de Automática,CBA 2002), Sao Paulo, Brazil, September 2006
33. Uzam M, Jones AH, Yucel I (2000) Using a Petri-net-based approach for the real-time supervisory control of an experimental manufacturing system. Int J Adv Manuf Technol 16(7):498–515
34. González PM (2001) GHENeSys: uma rede estendida orientada a objetos para o projeto de sistemas discretos. PhD dissertation, University of Sao Paulo, Brazil
35. Desel J, Esparza J (1995) Free choice Petri nets. Cambridge University Press, Cambridge, UK
36. Frey G, Schettler H-G (1998) Algebraic analysis of Petri Net based control algorithms. In: Proceedings of the 4th International Workshop on Discrete Event Systems (IEE WODES'98), Cagliari, Italy, August 1998, pp 94–96
37. Silva M (2000) Place transition II. Tutorial Petri nets 2000. In: Proceedings of the 21st International Conference on Application and Theory of Petri Nets, Aarhus, Denmark, June 2000
38. Benítez I (2002) Modelado modular para automatización de instalaciones docentes. In: Proceedings of the 2nd International Conference on Automatic Control (Automática, AUT 2002), Santiago de Cuba, Cuba, July 2002
39. Benítez I (2006) Modelagem de automação de sistemas de manufatura flexível com PN GHENeSys IEC61131. In: Proceedings of the 16th Latin-American Congress on Automatic Control (XVI Congresso Brasileiro de Automática,CBA 2006), Salvador, Bahia, Brazil, October 2006
40. Drath R (1997) A mathematical approach to describing a class of hybrid systems. In: Proceedings of the IEEE Workshop on Parallel and Distributed Real Time Systems, Geneva, Switzerland, April 1997