Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 1772–1783

Modeling, analysis and control of Discrete Event Systems: a Petri net perspective

Alessandro Giua[†], **Manuel Silva**[‡]

[†] *Aix Marseille Univ, Université de Toulon, CNRS, ENSAM, LSIS, Marseille, France (alessandro.giua@univ-amu.fr) and*
DIEE, University of Cagliari, Italy (giua@diee.unica.it)
[‡] *University of Zaragoza, Spain (silva@unizar.es)*

Abstract: The goal of this contribution is to briefly overview the historical development of the field of Petri nets under a System Theory and Automatic Control perspective. It is by far not meant to be comprehensive or inclusive, but to review through several representative areas a few of the conceptual issues studied in the literature. It was not possible to consider here the many domains of application were the Petri Nets modeling paradigm was used, among many others: manufacturing, logistic, hardware and software, protocols engineering, health management, transportation, etc.

1. PRELIMINARY OVERVIEW

Born in a Computer Science milieu, as Carl Adam Petri was fond of saying, nets belongs to *Systems Theory* in a broad sense. In the late fifties and beginning of the sixties of the past century, when the main focus was on local computations of mathematically intricate sequential problems, Petri developed a fresh approach to the conceptualization of *concurrency* and *synchronization*. In fact, the title of the seminal work of the field (Petri, 1962) is expressive: *Communication with Automata.*[1] Considering notions of *dependence* and *independence* of actions, *locality* of states and events were straightforwardly captured allowing *temporal realism* and *top-down* and *bottom-up* modeling approaches for concurrent-distributed Discrete Event Systems (DES).

Petri Nets (PNs) are bipartite valued graphs: *places* and *transitions* are the nodes and *weights* — inscriptions, more in general — are assigned to arcs. Their dynamics derives from the *marking* or distributed state.

At the beginning, PNs were only *autonomous*, meaning by that *untimed* or, more precisely, possessing only a *qualitative* notion of time: earlier or later; possibly at the same time. Also they were *non deterministic* models, a humble position leading to their logical study by contemplating all possible behaviors. The introduction of *quantitative* time dates to the middle of the seventies, when topics related to performance evaluation, verification and control, such as throughput computation, optimal scheduling, etc., started to be considered: Ramchandani (1973); Merlin (1974) and Sifakis (1977) are a small subset of representative early works on PN with time. In this sense PNs are *semi-interpreted*, i.e., there exist several "extended" or "interpreted" formalisms, suited to deal with diverse purposes but sharing the basic common principles. For example, beyond the many timed proposals, associating certain types of external events with the firing of transitions, *marking diagrams* (also *synchronized PNs*) constitute a clear generalizations of Moore or Mealy machines, in which the global state is substituted by a distributed one.

The above mentioned diversity of formalisms turns PNs into a conceptual framework or *paradigm* for the modeling of DEDS along their *life-cycle* (Silva and Teruel, 1996), allowing to deal with the formal representation and development of systems from preliminary design to performance evaluation and control, even including fault-tolerant implementation and operation. In particular, for a given system, this means to be able to check purely *logical* properties (such as boundedness, deadlock-freeness, liveness or reversibility in autonomous models), to compute *performance* properties (such as average values for: throughput of a subsystem; marking or queue length of a place; or utilization rate of a resource), to derive good *control* strategies (for example to minimize a make-span or to decide an optimal production mix), etc. In other words, a *modeling paradigm* is a conceptual framework that allows one to obtain modeling *formalisms* from some common concepts and principles with the consequent economy, coherence and synergy, among other benefits. As an example of synergy, we want to explicitly mention the computation of the *visit ratio* of transitions in an stochastic PN, allows to state some necessary or sufficient conditions for its liveness as autonomous. Campos et al. (1991) is the seminal work; a broader perspective of so called *rank theorems* is provided in Silva et al. (1998).

The first broad and organic perspective of works related to PNs is due to Brauer (1980). It integrates the "structural" line deriving from Petri first proposal and the "automata-language" based approach,[2] together with *Vector Addi-*

[⋆] This work has been partially supported by CICYT - FEDER project DPI2014-57252-R.

[1] For its translation into English, (Petri, 1966).

[2] Carl Adam Petri persistently claimed that formal languages (in the automata theory sense), were not appropriate to deal with the expressiveness of net systems models. In fact, their sequentialized views (sequences of events/occurrences of transitions) does not explicitly provide information about concurrency and distribution of the modeled system. Informally speaking, some kind of "isomor-

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2017.08.156

tion Systems (Karp and Miller, 1969) and other graphical models for parallel computations, independently introduced in the USA since the late sixties. From 1984 and for almost two decades, a significant part of the core of contributions to PN theory and applications was edited by Grezgorz Rozenberg as *Advances in Petri Nets*, a subseries of Lecture Notes in Computer Science (LNCS). Most of those contributions came from Informatics.

Although with different degree of centrality, the family of formalisms known as Petri Nets, have been considered in several disciplines, not only in Computer Science/Engineering (CSE), but also in Automatic Control (AC) and Operations Research (OR), with Mathematics and Logic always in the "back room" or "rearguard". Our focus in this work is mainly in the AC domain. Thus what is here presented is naturally a partial/biased view of the entire PN field.[3] The AC control community started discovering PNs in the second half of the seventies. For example, Moalla et al. (1980), following the spirit of the times, use them for modeling, verification, analysis and implementation of *logic controllers*.

Even if during the long period that has elapsed from 1962 an impressive number of results have been presented, a significant number of fundamental problems is still open. The impact of PNs on information technology can be assessed considering the conferences, courses, books, tools or standard norms (IEC, ISO, etc.) devoted to them. Applications of PN theory and methods exist in an extremely broad number of fields, among others: manufacturing, logistic, computer hardware and software, protocols engineering, traffic, biochemistry, population dynamics or epidemiology, for example.

In the eighties the quantitative timing of PNs generated a first "transient schism" (or divergence) in the PN community among those researchers accepting quantitative timed interpretations in PNs *versus* those rejecting them. Moreover, in the "combat" against the well-known *state-explosion problem* for DES, forms of *continuous* or *fluid* and *hybrid* PNs were introduced by the end of the eighties, what lead to some scientific controversy in the PN community of the times. The main argument against the new class of formalisms was that "real" PNs must be discrete models! In some sense, at the end of the past century and the beginning of the present one, this generated a second "transient schism" in the community among those researchers accepting particular fluid relaxations of PNs as "approximated" models for DES *versus* those rejecting them, somehow in parallel with the rising interest of the AC community in DESs. Even if we speak of "transients schisms", the modeling paradigm was always flexible enough to integrate the many "extensions" that do not contradict the basic concepts of PNs: bipartition, locality, consumption/production logic, etc.

This paper is structured as follows. In Section 2 the emergence of basic concepts is recalled and we are able to explicitly bring to the attention the family of PN formalisms as a modeling paradigm. Section 3 deals with the use of PNs as dynamical models to address classical problems of AC. Section 4 aims to create a bridge connnecting control theory and engineering of continuous, hybrid and discrete event systems. Finally a few promising areas that are open to future research are briefly discussed in Section 5.

2. PETRI NETS: FROM BASIC CONCEPTS TO THE MODELING PARADIGM

Due to space limitations, a very restricted subset of steps is traced in the sequel, starting with the seminal work of the field (Petri, 1962). In contrast with a widespread common vulgata, in this work there exists no PN in its classical graphical notation, something that appeared some three years later. In 2007 Petri confessed that "the graphical representation of structural knowledge which is now in widespread use I invented it in a playful mood in August 1939, and practiced it intensively for the purpose of memorizing chemical processes, using circles for substances and squares for reactions, interconnected by arrows to denote IN and OUT". The reason for this explicit omission was that he "did not want the theory to appear as a *graphical method* instead of a mathematical attack on the then prevailing Automata Theory, based on arguments taken from modern Physics".

The first net based formalism became what is known as Condition/Event nets, that are ordinary and 1-safe by definition. Its generalization to the more common Place/Transitions nets (PT-nets, most frequently simply denoted as PNs) happened during the second half of the sixties, appearing in the same years in the related works of the teams lead in the USA by Anatole Holt (working in private company) and by Jack B. Dennis (project MAC at MIT). Holt gave the name of "Petri Nets" to this class of formalisms. It was at this time that the fundamental differences between automata and PT-net systems (in the sequel simply PNs) were established. The most striking is the fact that while automata are characterized by a global symbolic state, in PNs the state is *distributed* and *numerical*. A place is a *local state variable* whose value (i.e., the *marking*) is a nonnegative integer, while a transition represents a *local event* whose occurrence changes the value of a subset of places. Moreover, the marking evolution logic is a non-monotonous *consumption/production logic* which straightforwardly allows the modeling of *unbounded* (non-finite) state spaces, and of the use of resources. As a consequence, *concurrency* (simultaneously enabled transitions that are not in *conflict*) and *synchronizations* (through *joins* or *rendez-vous*), can be naturally modeled. Therefore, stated from a different perspective, it can be said that *cooperation* and *competition* relationships can be directly represented.

The locality of places and transitions (and their *duality*) allows concurrent-distributed DES to be modeled interleaving in a free way *top-down* and *bottom-up* approaches. Differently stated, models can be constructed by *refining* transitions or places; also by *composing modules* through transitions (*synchronizations*) or through places (*fusions*), with the advantage that in any case the structure of modules is preserved.

---

phism" between the described system and the model contribute to the "faithfulness and understandability" of those formal constructions.

[3] For an historical perspective approaching a broader view on the development of the theory and its applications, together with elements of the development of the PN community, see (Silva, 2013).

During the first half of the seventies a second way of synchronization was integrated in PNs. Arcs were allowed to be labeled with non-negative integers *weights*, to describe the number of identical resources needed to fire a transition, or produced by its firing. These new nets were called "generalized" as opposed to the "ordinary" ones (by default, now a PN is a generalized one). Nevertheless, soon it was proved that generalized nets have the same "logical" expressive power of ordinary nets, although they may be more convenient from a modeling point of view. Moreover, Hack (1974) proved that *Vector Addition Systems*, ordinary and generalized PNs, and *Vector Replacement Systems* have the same expressive power.

From early times, there existed two alternative views concerning the development of the field. According to Peterson (1977), "in contrast to the work of Petri, Holt, and many European researchers, which emphasizes the fundamental concepts of systems, the work at MIT and many other American research centers concentrates on those mathematical aspects of Petri nets that are more closely related to automata theory [. . .] This mechanistic approach is quite different in orientation from the more philosophical approaches of Holt and Petri". In this sense, it is illuminating the pioneering comment by Holt and Commoner (1970) stating that, "perhaps we are closest in spirit to *operations research* techniques, but with an insistence on conceptual economy and rigor more common in purer branches of mathematics. Also, it is necessary that our descriptions be built up part by part in analogy to the way in which the systems being described are built up part by part". Formal languages or structural/compositional properties represent two different ways of addressing analysis and synthesis problems in the PN framework.

An important logical extension of PNs was the introduction of *inhibitor arcs* (Agerwala and Flynn, 1973), that allow the simulation of Turing Machines; soon this was followed by others similarly expressive extensions, such as *priority levels* on the firing of transitions. At the end of the seventies *High Level Petri Nets*, were introduced followed by more abstract (compact) formalisms, among which *Predicate/Transition Nets* and *Colored PNs* (Brauer, 1980; Jensen and Rozenberg, 1991). In high-level PN models tokens are *individualized* by means of labels (sometimes called *colors*). Information in tokens allows the objects to be named — i.e., they are no more indistinguishable — and dynamic associations can be created. Color sets are similar to *data types*, and are associated with places and transitions. Color *functions* inscribe the arcs connecting places to transitions and vice versa. The description of models using Colored PNs is at two levels: the "explicit" or high-level net structure (i.e., the basic relation of colored places and colored transitions), and the "implicit" structure, that is "hidden" in the functions attached to the arcs. If colors domains are finite, colored PNs constitute only a "modeling convenience" with no greater theoretical expressive power. The interest of colored PNs for dealing with manufacturing systems was recognized since the early eighties. For example, in Alla et al. (1985) part of a flexible workshop of the company Renault was modeled and validated. Colored nets have had an important impact in modeling industrial case studies in quite different domains (see, http://cs.au.dk/cpnets/industrial-use/). As an additional example, in (Dotoli and Fanti, 2006) timed Colored PNs have been used to model urban traffic.

Properties of PN models always depend on the net structure. They can be *behavioral*, if also depend on the initial marking, or *structural*, if the initial marking is abstracted. Among the first group are *reachability, boundedness, mutual exclusion, deadlock-freeness, liveness, home states,* etc. The abstraction of the initial marking can be done with the universal quantifier; for example, a PN is *structurally bounded* if it is bounded for any initial marking, or it is *structurally non-live* if it is non-live for any initial marking. Nevertheless, most frequently this last property is expressed as: a PN is *structurally live* if an initial marking exists such that the corresponding system is live.

During the seventies, the basis for three main analysis strategies of PN models were developed. While no one can offer a satisfactory solution for all cases of interest, in practice their combined use may be very effective. *Reachability* and *coverability graphs* deal with total or selective state enumeration. They are approaches in which "sequentialized views" are obtained, suffering thus from the state explosion problem. Moreover, the obtained graphs highly depend from the particular value of the initial marking. Among other developments to reduce the size of the state-space to be searched by a *model checking* algorithm, at late eighties-beginning of the nineties are the *stubborn sets* (a partial order technique) (Valmari, 1991) and the identification of *symmetries* (Starke, 1991); very recently in Ma et al. (2016) is proposed a compact representation of the reachability graph that uses the concept of *basis markings.* By keeping concurrency, *unfolding* techniques have the potential for reducing the computational complexity with respect to purely sequential enumeration, something better understood, for example, for 1-bounded systems (Giua and Xie, 2005; Esparza and Heljanko, 2008).

The complexity of Petri net decision procedures and the properties of PNs as *language generators* have been studied since the early 70s (Baker, 1973). Labels from an alphabet are assigned to transitions and, depending on the type of the labeling function and on the structure of the final marking set, a family of languages can be defined (Jantzen, 1991). We can think of the class of PN languages as a superset of regular languages and a subset of the class of context-sensitive languages. PNs are at the boundary between decidability and undecidability: in particular many problems are decidable only for deterministic PN languages (Vidal-Naquet, 1982).

Net *transformations* are *rewriting* techniques often exploited to reduce the net: in this sense they can be seen as structural approaches. The idea is to obtain models that are simpler to analyze while keeping the properties under study. If the transformation is of polynomial cost, but the analysis of the transformed system is exponentially cheaper, the advantage is obvious. For example, Berthelot (1986) showed that properties such as *boundedness*, *liveness* or the existence of *home states* can be analyzed by means of reduction rules involving *redundant places*, *pre-fusion* and *post-fusion.* Implicit places, i.e., places that are not the unique ones to prevent the firing of a transition, generalize redundant ones (Silva et al., 1998). However, the existence of irreducible net systems even for simple properties shows that the method, even if very interesting in practice, it is not complete.

Among the most original PN approaches are the so called *structural* techniques, that may be *graph-based* (using concepts as circuits, net components, siphons, traps, etc.) or *state-transition equation* based. In many works, the main idea is to consider subnets leading to some *invariants*: for example, a *P-semiflow*, a vector that is a non-negative left annuler of the incidence matrix, leads to a token conservation law and to a P-conservative component, i.e., a subnet. The use of siphons and traps (subsets of places), lead to some *stable predicates* and subnet components.

Looking for *invariant* properties, the approach based on the state-transition equation was introduced by Lautenbach and Schmid (1974). From a purely AC perspective, Murata (1977) presented the earliest contribution to the topic. Among early significant works highlighting the importance of *dual* views for the analysis of PN models based on places or on transitions are (Sifakis, 1978) and (Memmi and Roucairol, 1980); their importance resides in the fact that they bring together PNs and *convex geometry.* The systematic use of *linear programming* (with its duality, (un-)boundedness and convex geometry results) within PN theory was introduced by Silva and Colom (1988). In these settings, most frequently, only semi-decision algorithms are obtained because the solutions of the state-transition equation — that belong to the set of *nonnegative integers* for the firing count vectors — may be *spurious*, i.e., non-reachable in the PN system. Remarkably, the suitable addition of implicit places may remove spurious solutions (Colom and Silva, 1991) or can increase the Hamming distance between markings, i.e., by adding new places it is possible to increase the error-correcting capabilities of the implementation of the model.

When performance and performability evaluation is the goal, the net formalism should be extended by associating time with transitions (the most frequent option), places, arcs or tokens. Timing structures provide a means to reduce the non determinism of logical PNs by constraining the firing of transitions within time windows, stochastically (defining the pdfs, and probabilities at conflicts), or possibilistically (using fuzzy sets). Analytical techniques for stochastic models were inspired by previous developments within Queuing Network (QN) theory. They range from *exact* computations (e.g., Markov chains generation), through *approximations* (flow equivalent, or response time approaches, for example), to the computation of *bounds.* A distinctive point of PN theory is the extensive use of *net-driven* techniques (using structural decompositions, tensor algebra methods, symmetries, etc.). This subfield, which is still very active, started in the late 70s and reached its maturity by the end of the 90s (Ajmone Marsan et al., 1994; Balbo and Silva, 1998). Timed models are also used in real-time applications looking for correctness, from logical properties such as deadlock-freeness, to explicit response time-bounds.

Simulation of autonomous models refers to techniques to increase the confidence about correctness playing the “token game animation”, or looking for counterexamples or bugs, for example. Simulation of timed systems may be very helpful in practice, particularly if models are non Markovian. A key approach to *Discrete Event Simulation* was proposed in 1976 by Bernard Zeigler (see a companion paper in this session) where a “model-driven” perspective was introduced, leading to a separation of the model construction from the simulation techniques. PN-based simulation has always been essentially a model-driven approach.

Beyond analysis techniques, *implementation* issues are very important in building logic controllers. The easiness of translating PN formal models into executable code, allows not only *simulation* for correctness or performance analysis, but also *rapid prototyping* and *code generation*, possibly fault-tolerant. This topic was initially developed in the mid-seventies, in the area of *Programmable Logic Controllers* (PLCs) or general purpose computers, for example. The relations between PNs and Sequential Flow Charts, a graphical programming language developed from GRAFCET (defined in 1978), have often been explored.

Petri Nets of different *levels of abstraction* (C/E nets; nets; Predicate/Transition nets; Colored PNs; Object Oriented PN; etc.) and possibly enjoying a significant set of *interpreted extensions* (Marking Diagrams; Batches PNs; deterministic, stochastic or possibilistic-fuzzy timings, etc.) lead to a large “family” of formalisms. They can be used along the *life-cycle* of systems, allowing economy and coherence in modeling, analysis and control, also making possible synergies among the different tasks. Therefore, PNs constitute a broad *modeling paradigm*, constructed as a ”Cartesian product” of those corresponding to several levels of abstractions and of many extensions by interpretation (Silva and Teruel, 1996).

As a final comment, let us mention that a rich diversity of tools can be found on the Petri Nets World (2017) repository for the simulation and analysis of different Petri net models.

3. PETRI NETS AS DISCRETE EVENT MODELS FOR CONTROL SYSTEMS

In this section, we review the development of PN research within the area of DES, and show how they have been used to address classical problems of control systems in a broad sense and include analysis, control, diagnosis, state estimation and observability, identification, etc. A similar analysis concerning fluid PN models can be found in subsection 4.2.

DES have been formally considered in the framework of AC since the late 50s, being representative of such tradition the two *International Symposia on Discrete Systems* sponsored by IFAC in Riga (1974) and in Dresden (1977). Nevertheless, the modern area of DES within the AC community originated in the late 80s. In that period in the USA many DES researchers met at the Allerton Conference organized at the University of Illinois (one of longest-running conferences in the systems area) and a very first contribution dealing with supervisory control and Petri nets was presented by Krogh (1987). In Europe the first meeting of the WODES series (Workshop on Discrete Event Systems) was held in Prague in 1992. In the published proceeding of WODES'92 (Balemi et al., 1993) a few papers dealt with PNs: logical models were used for deadlock avoidance in flexible manufacturing systems or modeling supervisory control problems, while timed models were used for optimization and fluidisation. Since then the importance of PNs within the domain has increased.

3.1 Supervisory control

Supervisory control is a fundamental theory for the control of DESs that has been proposed in the 80s by Ramadge and Wonham (1989). This approach is very general and model independent: however, the original contributions and most of the subsequent developments focused on automata, that are intuitive models useful for presenting basic concepts. However, it is well known that automata have a smaller modeling power with respect to Petri nets: not only they can only describe finite state systems but they also lack explicit primitives to model important behavioral features such as concurrency and rendez-vous. Furthermore, they require the explicit enumeration of the state space and lack computationally efficient algorithms for analysis and synthesis. For this reason, Petri nets have been considered as suitable model for supervisory design since the very beginning, with the dual objective of enlarging the class of control problems considered and of exploiting the many algebraic analysis techniques that pertain to them. An early review of this area of research was presented by Holloway et al. (1997).

The original paradigm of supervisory control is concerned with language specifications, i.e., the desired behaviour of the plant under control is expressed as a set of legal event sequences it should generate. The theory is quite general but most of the presented results consider automata as discrete event models and, correspondingly, concern regular languages. However, PNs soon started to be used within this framework: as an example, Krogh (1987) showed that under concurrent firing of transitions a supremal controllable language may not exist. While PN models allow one to extend the classes of systems and specifications considered, including also infinite state systems, supervisory control problems have been shown to be undecidable for arbitrary PNs (Giua and DiCesare, 1994). This undesirable feature can be avoided by restricting the class of models considered to deterministic PNs, whose language class is still a proper superset of the class of regular language. The standard approach for control with language specifications requires in a first step to construct the parallel composition of the net describing the plant with the net describing the specification. This step is very efficient using PNs (polynomial in the size of the nets) and has an additional nice feature: the overall model represents the closed-loop system where one can clearly distinguish the original plant and the specification structure, than can be seen as the controller. Unfortunately, due to the presence of uncontrollable transitions, one needs to refine this structure to avoid reaching undesirable — e.g., uncontrollable or blocking — markings. The set of undesirable markings usually does not have a special structure and it is not obvious how the net can be refined to prevent reaching them. A general approach that can be used to refine a net is based on the *theory of regions* (Badouel and Darondeau, 1998) that will be briefly discussed in subsection 3.3: it can be used to design maximally permissive controllers but it requires an exhaustive enumeration of the state space and the required additional control structure can be very large, as big as the set of markings to forbid. For this reason, we believe that the use of PNs in supervisory control for language specifications is even today an open area of research, where efficient techniques are still missing.

Another very active subdomain of supervisory control deals with the control objective of preventing a plant from reaching a set of undesirable markings. This is a problem that can be addressed leveraging the many features that directly pertain to the PN models, including the fact that the state (marking) is represented by a vector and that the knowledge of the net structure may often be enough to characterize its evolution. A first approach was presented by Holloway and Krogh (1990) showing that it was possible to efficiently solve supervisory control problems that require preventing some places from getting marked by the analysis of the uncontrollable paths that lead from a given controllable transition to the place itself.

A large number of works deal with the control synthesis for a wider class of state specifications called *Generalized Mutual Exclusion Constraints* (GMECs) which define a convex set of legal markings and were introduced by Giua et al. (1992). The main advantage of these constraints is the fact that they can be enforced on a PN by simple control structures called *monitor places* whose design is based on the net structure and does not require explore the reachability space. An additional interesting feature is the fact that the net representing the plant with the addition of the controller (the monitor) describes the *closed-loop system*. One of the drawback of this approach is the fact that when the net contains uncontrollable or unobservable transitions a maximally permissive solution may be difficult to compute (requiring at least a partial reachability analysis) and may not always be enforceable by monitor places. The original solution proposed by Moody and Antsaklis (2000) consists of an elegant algebraic procedure to design a *suboptimal* monitor place, i.e., a monitor that solves the original control problem but may not be the maximally permissive supervisor. Many subsequent developments followed, including the extension of this approach to more general constraints involving firing vectors (Iordache and Antsaklis, 2003), to constraint transformation rules that ensure maximally permissiveness for subclasses of nets (Luo and Nonami, 2011) and finally to non-convex legal sets (Ma et al., 2015).

3.2 Deadlock and liveness analysis and control

Deadlock freeness and *liveness* are basic properties that have, in the DES domain, an importance comparable to that of stability for time-driven systems. A deadlock represents an anomalous state from which no further evolution is possible. Liveness is a stronger property, requiring that from all reachable state any transition firing can *eventually* occur. They are relevant issues in many automation problems and appropriate strategies should be occasionally adopted in order to enforce them.

Deadlock (and partially also liveness) analysis for PNs based on structural approaches has been mostly addressed under the assumption that all transitions are controllable. For ordinary nets, if a deadlock is reached, the set of unmarked places defines a *siphon* (or *structural deadlock*).

Thus, if no minimal siphon can be emptied (because it contains a marked trap) the system is *deadlock-free* (DF) (Commoner, 1972). For some net subclasses, such as Free-Choice nets, DF implies liveness. Nevertheless, the siphon/trap condition is computationally hard to check. Chu and Xie (1997) formulate a deadlock detection method by solving integer linear programming such that either a complete siphon or state enumeration is not necessary.

Concerning liveness analysis, since the early 90s the main focus has been on efficient structural approaches for the case that all transitions are controllable. We mention here an approach, initially suggested by the computation of *visit ratios* in stochastic PNs, leading to the so called *rank theorems*, i.e., a family of results about necessary or sufficient conditions for structural liveness of structurally bounded nets (for particular net subclasses, such as Free-Choice, it lead to a necessary and sufficient condition for liveness). Their computation is in polynomial time For a presentation of both linear algebra based analysis techniques and other results as the elimination of some spurious solutions —that can also be used for control— see (Silva et al., 1998).

What to do if a given PN deadlocks or is not live? How to constrain its behavior in such a way that the system becomes DF or live?

Concerning deadlock control, a first distinction is between *deadlock avoidance* and *deadlock prevention* (Viswanadham et al., 1990). Approaches of the first type look for on-line control policies that avoid reaching a deadlock state, while approaches of second type modify the net structure by adding a suitable control structure, to ensure deadlock-freeness for the closed-loop system. A seminal contribution for preventing deadlocks in flexible manufacturing systems was presented by Ezpeleta et al. (1995): the basic idea is to control all strict minimal siphons imposing a GMEC for each siphon to prevent it from becoming unmarked. Unfortunately the number of such siphons may be exponential in the net size and the method only applies to a restricted net class. The search of more efficient solutions or the generalization of the net subclasses to which the computations apply is a very active topic in PN theory. Park and Reveliotis (2001); Ezpeleta et al. (2002); Li and Zhou (2004) are a few of the relevant works in the domain.

The previous approaches are efficient but not necessarily maximally permissive: in order to find a maximally permissive deadlock-free controller the full reachability set has to be generated to ensure that the supervisor disables at some particular markings the transitions whose firing leads a system from the safe marking set to the unsafe space as in (Uzam, 2002). As a further improvement of this approach, Chen and Li (2011) show that only a minimal sets of safe and unsafe markings need to be considered, leading to a reduced computational overhead and also to a simpler control structure.

Liveness enforcing by supervisory control has also been explored, but this is a complex problem especially when some of the transitions of the net are not controllable (Sreenivas, 1997). Efficient solutions to this problem are still missing.

3.3 State estimation, diagnosis and identification

Control theory has considered several interesting problems that are based on the (partial) observation of a system's behavior.

The *state estimation problem* consists in reconstructing the current and past values of a system's state from the knowledge of the current and past values of its external measurable outputs. Using Petri nets one usually assumes that measurable outputs are labels assigned to transitions that fire but often the token content of some places is also assumed to be measurable (Ramirez-Trevino et al., 2003). A different setting, more similar to the state estimation of time-driven setting, has also been explored by Giua and Seatzu (2002): assuming that the initial state is unknown but all transition firings are observable, one can construct an observer whose estimation error hopefully goes to zero. More recently these approaches have also been extended to timed (Declerck and Bonhomme, 2014) or time nets (Basile et al., 2015).

Another classical problem is that of *fault analysis* or *diagnosis* of a dynamical system, i.e., detecting the occurrence of a fault. A few fault detection methods based on PN models were developed in the 80s and early 90s. They included combining error detection/correction codes to represent the marking (its *Hamming distance* is increased by adding some redundancies) and reducing the PN model while preserving the subset of *observable transitions* (Velilla and Silva, 1988). Other approaches were based on monitoring the tokens in P-invariants (Prock, 1991) or backfiring transitions to determine if a given state is invalid (Sreenivas and Jafari, 1993). Years later, Benveniste et al. (2003) used *net unfolding* to avoid generating the full reachability space of a system to detect if a given fault pattern has occurred.

Subsequent approaches were inspired by the theory developed by Lafortune and co-authors (Sampath et al., 1995) and also dealt with the diagnosability analysis of a given system, i.e., determining if the occurrence of a fault can be detected. The use of PNs in this area has been primarily motivated by the need of practically reducing the computational complexity of solving a diagnosis or diagnosability problem. It should be noted that a common assumption is that both the *nominal model* and the *fault model* of the system is given. Usually faults are modeled by unobservable event whose occurrence must be detected based on the system's observation: thus it is not surprising that almost all approaches exploit techniques previously developed for state estimation. Among the PN techniques used in this setting we recall the notions of border places to partition a net in simpler subnets to analyze separately (Genc and Lafortune, 2007), *minimal explanations* (Jiroveanu and Boel, 2010) and *basis markings* (Cabasino et al., 2010) that try to avoid the full construction of the reachability set, or on-line approaches based on integer programming (Basile et al., 2009; Dotoli et al., 2009). PNs also allow the extension of this approach to infinite state systems (Cabasino et al., 2012) and to time nets (Basile et al., 2015). In recent years some of these approaches are also being used to study *opacity*, i.e., the property of a system to keep its data private from an intruder that can partially observe its evolution (Bryans et al., February, 2005): this a topic of great relevance in the context of multi-agent systems, internet of things, etc.

We also mention the classical problem of *identification*, i.e., building a mathematical models of a dynamical system from measured data. In the domain of PNs, identification is also known with the name of *net synthesis* (Badouel et al., 2015) and is closely related to the process mining technique called *discovery*, i.e., constructing an unknown model based on an event log (Van Dongen et al., 2009).

Process mining approaches usually consider large amount of data and aim to derive in an efficient way a partial approximate model. On the contrary, net synthesis approaches look for an exact model and as such they have a very high complexity. A first technique for net synthesis is based on the *theory of regions*: it synthesizes a Petri net from a transition system adding places to make sure counterexamples (i.e., sequences not belonging to the transition system) cannot occur on the net (Badouel and Darondeau, 1998). Other approaches still consider a list of examples and counterexamples but determine the net structure by solving an Integer Programming problem whose complexity can grow very large: these procedure can be applied either off-line (Cabasino et al., 2007) or on-line (Dotoli et al., 2008). Finally the identification of fault models has also been addressed (Wu and Hadjicostis, 2005).

4. FLUIDIZATION OF PN MODELS AND FLUID VIEWS OF SYSTEMS

The *state explosion problem* that characterizes DES poses strong limitation to analysis and synthesis methods for all formalisms with reasonable modeling capabilities such as Queuing Networks (QNs), Petri Nets (PNs) or, more recently, Process Algebras (PAs) (Silva et al., 2011). Fluid or continuous PNs are obtained by a simple relaxation. The underlying idea is not really new (remember the Lotka-Volterra equations, 1926-27), and has been —explicitly or implicitly— employed in application domains such as manufacturing, communication or transportation systems; also in populations dynamics problems, in fields as Biology, Ecology or Epidemiology. Fluid models “over approximate” the set of reachable states (markings in the PNs case) of their discrete counterpart.

In fluid PN models, the firing amount of the transitions are relaxed to non-negative real quantities. The introduction of fluidization in the Petri net paradigm dates back to 1987 (David and Alla, 1987). As explicitly stated by David and Alla (2010), the source of inspiration was the fluidization of models for the performance evaluation of production lines (manufacturing domain). At the same meeting in Zaragoza, working with the *fundamental* or *state-transition equation* of the PN system, the systematic use of linear programming techniques for the structural analysis of PNs was proposed by Silva and Colom (1988). This second approach can be simply “rephrased” as relaxing Integer Programming into Linear Programming in order to obtain: necessary or sufficient conditions for qualitative properties (such as boundedness or deadlock-freeness, for example); or bounds for quantitative ones (on the marking of a place in an untimed model, or of the throughput of a transition in a timed model, for example). Fluid QNs are intrinsically timed, but models based on PNs (and PAs) can be *untimed* or *timed*. Therefore, fluidization of PNs has been historically considered at *logical* and at *performance* levels.

In the study of fluid PN models, properties of discrete nets such as deadlock-freeness, boundedness, observability or controllability, are similarly of primary interest. If the continuous model is an approximation of a discrete one, major concerns are the understanding of the validity or accuracy of the approximation, and finding a good trade-off with respect to computational complexity and decidability issues. It can be said that with fluidizable PNs (we stress that not all are!), the bigger the initial marking (or population), the better. In fact, a double advantage exists: greater accuracy (smaller relative error in timed models, for example) and more importantly computational savings (exponential decrease on those efforts). From a historical perspective, most of the works during the first period (till the mid of the first decade of this century) focus primarily on exploring the potentialities for the analysis of the new classes of models, while topics such as their *improvement* and *legitimization* were addressed years later. Silva (2016) provides a recent perspective on the fluidization of PNs.

4.1 On the fluidization of DES and fluid views

As the linearization of a continuous dynamical system, the fluidization of a PN model is a relaxation that has to be used with care, even when untimed PN systems are considered. Even the simplest models can be affected by the classical “Zeno paradox” which leads to the idea of *reachability at the limit* (lim-reachability). In other words, even if the PN system is bounded, infinite sequences should be considered, which may lead to behaviors that are not possible in discrete models, such a the emptying of traps (Recalde et al., 1999). Moreover, fluidization cannot always be applied because significant discrepancies between continuous and discrete behaviors may appear. Fluidization and linearization are two complementary relaxations, the second not being always applicable; for example, if the system is chaotic. In the present context, deadlock-freeness of the untimed discrete model may be neither necessary nor sufficient for the corresponding fluid one. Those discrepancies can be formally studied through the concept of *marking homothetic monotonicity* (weaker than the more classical marking monotonicity) in the discrete model (Fraca et al., 2014).

Depending on the time interpretation of the discrete model and the net structure, *Timed Continuous PN* (TCPN) can be defined in many different ways, particularly when dealing with *rendez-vous*, a syncronization primitive that can be modeled by means of transitions with more than one input arc. Two basic timing interpretations are *constant* and *variable speed*, also known as *finite server semantics* (FSS) and *infinite server semantics* (ISS), respectively. Fluid or continuous TCPN under FSS or under ISS are “technically” *time-driven hybrid* systems.

For a comprehensive discussion of FSS-TCPN, see David and Alla (2010), while many results concerning ISS-TCPN are summarized in Silva et al. (2011). ISS is most frequently used in the context of manufacturing, logistic or hospital management problems. Moreover, it has been proved that for particular net systems (for example, mono-T-semiflow (MTS) nets under some general conditions), ISS approximates better the steady-state flows (Mahulea et al., 2009). By using ISS, basic properties of the discrete Markovian PN model are inherited, and dynamic TCPN systems represent a particular class of *piecewise affine systems with a polytopic partition* in which the derivatives of the marking are continuous functions.

Nevertheless, the equilibrium markings (i.e., the null solutions of the state-equation) may be *non-monotonic* w.r.t. the *initial marking* (i.e., more resources may reduce the throughput) or w.r.t. the *firing rates* associated with transitions (i.e., faster machines may lead to a smaller throughput). Moreover, those non-monotonicities frequently coexist with *discontinuities* for steady-state behaviors (Meyer, 2012). Monotonicity w.r.t. the marking has been very recently characterized in pure structural terms for some broad class of nets; moreover, if the system is monotonic with respect to the marking, no discontinuity may appear in the steady-state throughput. Additionally, TCPN under ISS may simulate *Turing machines*, thus they have an important theoretical expressive power; the reverse of the coin is that some properties such as marking coverability, submarking reachability or the existence of a steady-state may remain *undecidable.*

Decolorizing high-level PNs (such as colored PNs), the *minimum* operator of ISS may become a *product.* This lead to so called *population* or *product semantics* (Silva et al., 2011), very frequently used in System Dynamics, a class of time-driven fluid PNs that are not hybrid systems. Being possible to define firing flows proportional to the product of the marking of input places, *chaotic* models may be easily described.

As discrepancies between fluid and discrete behaviors may appear, a key question is *how to improve fluid approximations?* Several techniques have been developed: while they alleviate the problem in many practical cases, it should be pointed out that they do not solve it in general! Among other improvement techniques (Silva, 2016): (1) The use of *cutting implicit places* to remove *spurious solutions* (for example, spurious deadlocks); (2) The introduction of variations in the ISS to take into account the *weighted arcs* from places to transitions, firing constraints that are “not seen” under ISS, because markings are assumed to be very large; and (3) The addition of *noise* to the firing flows, what lead to *stochastic* continuous approximations. If the marking is “relatively small”, the removing of spurious solutions and the so called *rho-semantics* may be of great interest (Fraca et al., 2017). Stochastic approximations (Vázquez and Silva, 2012; Beccuti et al., 2014) may be particularly interesting if the trajectory of the system frequently crosses the border of regions of the polytopic partition of the reachable space.

Besides providing improvements, the previous results contribute to *legitimize* the continuous relaxation of the corresponding discrete model. Another server semantics (i.e., timing interpretation) for TCPN is proposed by Lefebvre et al. (2010).

4.2 On the use of fluid models

Since the early works it was clear that fluid PNs enjoy important advantages (Recalde et al., 1999). Unfortunately, some modeling features such as mutual exclusion relationships cannot be observed in continuous systems, since they are based on the notion of *disjunctive resources.* The same can be said for particular *monopoly* and *fairness* situations, among others.

In analogy with (discrete) PNs, their continuous counterparts can be analyzed using *transformation* and *structural* techniques, but not *reachability enumeration*, unless the reachability space is discretized into zones. *Model checking* techniques deal with formal verification of DESs in the latter case. For a TCPN system under ISS, formal analysis may start by embedding it into a *Piece-Wise Affine* (PWA) system and, by means of discrete abstractions, into a *finite transition system* (Kloetzer et al., 2010). Genuine to this paradigm, structural techniques allows to efficiently study necessary or sufficient conditions for many interesting properties. The join use of net transformations and structural techniques, together with simulation is most frequently very interesting in practice.

Let us now briefly focus on classical control theory properties, such as observation, diagnostic and control. In particular, a blend of techniques belonging to PNs and (continuous and hybrid) Automatic Control is used, emphasizing some structural (graph and algebraic) concepts and results.

Assuming that the marking of a subset of places (or the flow of a subset of transitions) can be observed (measured), a TCPN system under ISS is said to be *observable* if it is always possible to compute its initial marking. In many real cases, the possibility to estimate/observe the system for all possible values of the firing rates is an important issue. A stronger property that only depends on the net structure, regardless of the firing rates associated with transitions, is *structural observability.* It can be approached using graph-based arguments. Moreover, a TCPN is said to be *generically observable* if it is always observable, outside of a proper algebraic variety of the firing rates space (Silva et al., 2011). Lefebvre (2001) deal with discrete-time models and measure some places, the goal being to estimate the firing flows of the transitions.

Related to observability, is the problem of *fault diagnosis*, Using the characterization of the set of consistent markings and the algorithm to compute it, the problem of fault detection for systems modeled by untimed CPN has been addressed in (Mahulea et al., 2012). The main advantage of fluidification is that more general Petri net structures than those taken into account in discrete approaches can be considered (in particular, the unobservable subnet needs not to be acyclic).

Looking at the optimization of systems, not only *mathematical programming* methods can be applied, for example to compute: an optimal initial marking, an optimal routing rate at specific conflicts, or an optimal steady-state; etc. (see, for example, chapter 18 in Seatzu et al. (2013)). In Wardi et al. (2013) the *Infinitesimal Perturbation Analysis* (IPA), a gradient-estimation technique, is extended from stochastic flow models to stochastic Marked Graphs.

Control objectives in DES may be to "enforce" some safety specifications (e.g. deadlock-freeness or particular mutual exclusion constraints). This can only be done by reducing the firing flow of selected transitions at particular states, The key point is that any control action allowed in the TCPN system may only slow down the nominal or uncontrolled flow, since transitions —machines for instance— cannot work faster than their nominal speed. In discrete PNs, such control action is equivalent to temporarily blocking the firing of enabled transitions. Otherwise stated, control actions only can reduce the flow through specific transitions, those named *controllable.* If all the transitions are controllable, controllability at the net level has a very simple structural characterization, *consistency*; otherwise, the controllability criteria is much more intricate (Vázquez et al., 2014). Similarly to observability, generic and structural controllability are research goals, but further work is necessary. In the literature many works deal with the computation of controllers (fuzzy, linear matrix inequalities, ON/OFF, model predictive control, decentralized, distributed, etc.), most of them dealing with systems in which all transitions are controllable (see, for example, Silva et al. (2011), and chapter 20 in Seatzu et al. (2013)). Among problems that did not receive yet satisfactory solution is the decentralized control of ISS-TCPN with uncontrollable transitions, a problem of relevance for systems of large dimension.

4.3 A brief perspective on hybrid PNs

The extension from (discrete) net models to *Hybrid PNs* (HPNs) did follow many complementary lines. A basic one is by relaxing the firing of a *subset* of transitions, therefore, discrete and continuous transitions are mixed (Bail et al., 1991; Trivedi and Kulkarni, 1993). Most works on HPNs deal with timed formalisms, nevertheless, untimed models and analysis techniques are also of interest. We should warn that, as was the case for continuous nets, the fluidization of a single transition of a PN may transform, say, a live system into a dead-lockable one.

An alternative basic approach to define hybrid PN formalisms, derives from *hybrid automata* (Alur and Dill, 1994); the main idea is to keep a discrete PN model (to describe the event-driven dynamics) while adding continuous variables governed by algebraic or differential equations; in other words, these approaches extend the classic PN formalism by a continuous time-driven interpretation (see, for example, Champagnat et al. (1998)).

During the 90s of the past century many hybrid PN proposals appeared. By the end of the decade, in the broader framework of hybrid systems, several of those are considered in Antsaklis et al. (1998); at the very beginning of the new century, a survey centered on hybrid PNs by Di Febbraro, A. Giua and G. Menga (eds.) (2001) provides a more detailed landscape. *Batches PNs* (Demongodin, 2001; Demongodin and Giua, 2010) or *First-Order Hybrid PNs* (FOHPNs) (Balduzzi et al., 2000) are complementary hybrid extensions, appropriate for different modeling problems. Of additional interest is the special issue by Cassandras et al. (2008).

Several approaches have been used for optimization of hybrid nets. Mathematical programming methods have often be applied to compute: optimal initial markings or optimal transitions firings (Balduzzi et al., 2000); also optimal routing rates at conflicts (Gaujal and Giua, 2004).

5. SOME OPEN PROBLEMS

We conclude this historical perspective by suggesting a few areas that are open for future research.

PNs have been a valuable model for supervisory control of discrete event systems and can reduce the complexity of supervisory synthesis. However, while the use of Petri nets with state specifications is a very mature area (consider as an example the works on GMECs), their use in the design of controllers for more general behavioral specifications has not been equally successful: finding a general approach based on structural analysis to address the latter issue is still an open problem. In addition, we believe that the supervisory control for timed systems could benefit from the use of PN models with an implicit dense time interpretation, as opposed to an explicit discretization of clock events that produces unnecessary complex models.

Colored PNs have been used to model large systems characterized by partial symmetries and some structural approaches for their analysis have been developed. What has not received much attention so far is the use of this class of nets for solving problems of supervisory control, state estimation, diagnosis and identification. This is a potentially fruitful area that deserves to be explored.

Control theory concepts have been applied to fluid and hybrid PNs in the last years. We believe that many control and observation techniques recently developed can be applied in this context and new interesting results are expected.

PNs provide a natural way of describing distributed systems, due to the inherently local representation of states and events. The control community has seen a recent surge of interest in the area of networked control systems usually modelled as time-driven systems although in recent developments, such as event-based control (Grüne et al., 2014), the advantages of event triggered communications have been explored. It is likely that in the immediate future the study of networked discrete event systems will see a parallel growth where PNs may play an important role.

REFERENCES

T. Agerwala and M. Flynn. Comments on Capabilities, Limitations and "Correctness" of Petri Nets. *ACM SIGARCH Comp. Architecture News*, 2(4):81–86, 1973.

M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets.* John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994. ISBN 0471930598.

H. Alla, P. Ladet, J. Martínez, and M. Silva. Modelling and Validation of Complex Systems modeled by Colored Petri Nets. In G. Rozenberg, editor, *Advances in Petri Nets 1984*, volume 609 of LNCS, pages 15–31. Springer-Verlag, 1985.

R. Alur and D.L. Dill. The theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

P. J. Antsaklis, X.D. Koutsouko, and J. Zaytoon. On Hybrid Control of Complex Systems: A Survey. *Eu-*

ropean Journal of Automation (APII-JESA), 32(9-10): 1023–1045, 1998.

E. Badouel and Ph. Darondeau. Theory of regions. In *Lectures on Petri Nets I: Basic Models*, pages 529–586. Springer, 1998.

E. Badouel, L. Bernardinello, and P. Darondeau. *Petri Net Synthesis*. Springer, 2015.

J. Le Bail, H. Alla, and R. David. Hybrid Petri nets. In *Proc. 1st Int. European Control Conference (ECC91)*, pages 1472–1477, Grenoble, France, 1991.

H.G. Baker. Equivalence problems in Petri nets. Technical Report Computation Structures Group. Project MAC, Lab. for Computer Science, M.I.T., Cambridge, MA, 1973.

G. Balbo and M. Silva, editors. *Performance Models for Discrete Event Systems with Synchronizations: Formalisms and Analysis Techniques.* MATCH Performance Advanced School, Jaca, Spain, 1998. (https:webdiis.unizar.es/GISED/?q=news/matchbook).

F. Balduzzi, A. Giua, and G. Menga. First-Order Hybrid Petri nets: A model for optimization and control. *IEEE Trans. on Robotics and Automation*, 16(4):382–399, 2000.

S. Balemi, P. Kozák, and R. Smedinga, editors. *Discrete Event Systems: Modeling and Control*, volume 13 of *Progress in Systems and Control Theory.* Birkhaüsen, 1993.

F. Basile, P. Chiacchio, and G. De Tommasi. An efficient approach for online diagnosis of discrete event systems. *IEEE Trans. on Automatic Control*, 54(4):748–759, April 2009.

F. Basile, M.P. Cabasino, and C. Seatzu. State estimation and fault diagnosis of labeled time Petri net systems with unobservable transitions. *IEEE Trans. on Automatic Control*, 60(4):997–1009, April 2015.

M. Beccuti, E. Bibbona, A. Horváth, R. Sirovich, A. Angius, and G. Balbo. Analysis of Petri Net Models through Stochastic Differential Equations. In G. Ciardo and E. Kindler, editors, *Application and Theory of Petri Nets and Concurrency*, volume 8489 of *LNCS*, pages 273–293. Springer, 2014.

A. Benveniste, E. Fabre, S. Haar, and C. Jard. Diagnosis of asynchronous discrete-event systems: a net unfolding approach. *IEEE Trans. on Automatic Control*, 48(5): 714–727, May 2003.

G. Berthelot. Checking Properties of Nets Using Transformations. In G. Rozenberg, editor, *Advances in Petri Nets 1985*, volume 222 of *LNCS*, pages 19–40. Springer, 1986.

W. Brauer, editor. *Net Theory and Applications*, volume 84 of *LNCS*. Springer, 1980.

J. W. Bryans, M. Koutny, and P. Y. A. Ryan. Modelling opacity using Petri nets. *Electron. Notes Theor. Comput. Sci.*, 121(4):101–115, February, 2005.

M.P. Cabasino, A. Giua, and C. Seatzu. Identification of Petri nets from knowledge of their language. *Discrete Event Dynamic Systems*, 17(4):447–474, 2007.

M.P. Cabasino, A. Giua, and C. Seatzu. Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica*, 46(9):1531–1539, 2010.

M.P. Cabasino, A. Giua, S. Lafortune, and C. Seatzu. A new approach for diagnosability analysis of Petri nets using verifier nets. *IEEE Trans. on Automatic Control*, 57(12):3104–3117, Dec 2012.

J. Campos, G. Chiola, and M. Silva. Properties and Performance Bounds for Closed Free Choice Synchronized Monoclass Queueing Networks. *IEEE Trans. on Automatic Control*, 36(12):1368–1382, 1991.

C. G. Cassandras, A. Giua, C. Seatzu, and J. Zaytoon (eds.). Special Issue on Discrete Event Methodologies for Hybrid Systems. *Discrete Event Dynamic Systems*, 18(2), 2008.

R. Champagnat, P. Esteban, H. Pingaud, and R. Valette. Petri net based modeling of hybrid systems. *Computers in Industry*, 36(1-2):139–146, 1998.

Y.F. Chen and Z.W. Li. Design of a maximally permissive liveness-enforcing supervisor with a compressed supervisory structure for flexible manufacturing systems. *Automatica*, 47(5):1028–1034, 2011.

F. Chu and X.L. Xie. Deadlock analysis of Petri nets using siphons and mathematical programming. *IEEE Trans. on Robotics and Automation*, 13(6):793–804, Dec 1997.

J.M. Colom and M. Silva. Improving the linearly based characterization of P/T nets. In G. Rozenberg, editor, *Advances in Petri Nets'90*, volume 483 of *LNCS*, pages 113–145. Springer-Verlag, 1991.

F. Commoner. Deadlocks in Petri nets. Technical Report Report CA-7206-2311, Laboratory for Computer Science, M.I.T., Wakefield, MA, 1972.

R. David and H. Alla. Continuous Petri Nets. In *Proc. of the 8th European Workshop on Application and Theory of Petri Nets*, pages 275–294, Zaragoza, 1987.

R. David and H. Alla. *Discrete, Continuous and Hybrid Petri Nets*. Springer, 2010. (1^{st} ed., 2004).

P. Declerck and P. Bonhomme. State estimation of timed labeled Petri nets with unobservable transitions. *IEEE Trans. on Automation Science and Engineering*, 11(1): 103–110, Jan 2014.

I. Demongodin. Generalised Batches Petri Net: Hybrid model for high speed systems with variable delays. *Discrete Event Dynamic Systems*, 11(1-2):137–162, 2001.

I. Demongodin and A. Giua. Linear programming techniques for analysis and control of Batches Petri nets. In *Procs. of the Int. Workshop on Discrete Event Systems, WODES'10*, pages 4–9, 2010.

A. Di Febbraro, A. Giua and G. Menga (eds.). Special Issue on Hybrid Petri Nets. *Discrete Event Dynamic Systems*, 11(1-2), 2001.

M. Dotoli and M.P. Fanti. An urban traffic network model via coloured timed Petri nets. *Control Engineering Practice*, 14:1213?1229, 2006.

M. Dotoli, M.P. Fanti, and A.M. Mangini. Real time identification of discrete event systems using Petri nets. *Automatica*, 44(5):1209–1219, 2008.

M. Dotoli, M.P. Fanti, A.M. Mangini, and W. Ukovich. On-line fault detection in discrete event systems by Petri nets and integer linear programming. *Automatica*, 45 (11):2665–2672, 2009.

J. Esparza and K. Heljanko. *Unfoldings ? A Partial-Order Approach to Model Checking.* EATCS Monographs in Theoretical Computer Science. Springer-Verlag, 1st edition, 2008.

J. Ezpeleta, J. M. Colom, and J. Martínez. A Petri Net Based Deadlock Prevention Policy for Flexible Manufacturing Systems. *IEEE Trans. on Robotics and Automation*, 11(2):173–184, 1995.

J. Ezpeleta, F. Tricas, F. García-Vallés, and J.M. Colom. A banker's solution for deadlock avoidance in FMS with flexible routing and multiresource states. *IEEE Trans. on Robotics and Automation*, 18:621–625, August 2002. ISSN 1042-296X.

E. Fraca, J. Júlvez, and M. Silva. On the fluidization of Petri nets and marking homothecy. *Nonlinear Analysis: Hybrid Systems*, 12(2):3–19, 2014.

E. Fraca, J. Júlvez, and M. Silva. Fluid approximation of Petri net models with relatively small populations. *Discrete Event Dynamic Systems*, 14, 2017. doi: 10.1007/s10626-017-0238-9.

B. Gaujal and A. Giua. Optimal stationary behavior for a class of timed continuous Petri nets. *Automatica*, 40 (9):1505 – 1516, 2004.

S. Genc and S. Lafortune. Distributed diagnosis of place-bordered Petri nets. *IEEE Trans. on Automation Science and Engineering*, 4(2):206–219, April 2007.

A. Giua and F. DiCesare. Blocking and controllability of Petri nets in supervisory control. *IEEE Trans. on Automatic Control*, 39(4):818–823, Apr 1994.

A. Giua and C. Seatzu. Observability of place/transition nets. *IEEE Trans. on Automatic Control*, 47(9):1424–1437, Sep 2002.

A. Giua and X. Xie. Control of safe ordinary Petri nets using unfolding. *Discrete Event Dynamic Systems*, 15: 349–373, 2005. doi: 10.1007/s10626-017-0238-9.

A. Giua, F. DiCesare, and M. Silva. Generalized Mutual Exclusion Constraints on Nets with Uncontrollable Transitions. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 974–979, 1992.

L. Grüne, S. Hirche, O. Junge, P. Koltai, D. Lehmann, J. Lunze, A. Molin, R. Sailer, M. Sigurani, C. Stöcker, and F. Wirth. Event-based control. In J. Lunze, editor, *Control Theory of Digitally Networked Dynamic Systems*, pages 169–261. Springer, 2014.

M. H. T. Hack. Decision problems for Petri nets and Vector Addition Systems. Technical Report Computation Structures Group Memo 95. Project MAC, Laboratory for Computer Science, M.I.T., Cambridge, MA, 1974.

L.E. Holloway and B.H. Krogh. Synthesis of feedback control logic for a class of controlled Petri nets. *IEEE Trans. on Automatic Control*, 35(5):514–523, May 1990.

L.E. Holloway, B.H. Krogh, and A. Giua. A survey of Petri net methods for controlled discrete event systems. *Discrete Event Dynamic Systems*, 7(2):151–190, 1997.

A. W. Holt and F. Commoner. Events and Conditions. *Record Project MAC Conference on Concurrent Systems Parallel Computation*, pages 3–52, 1970.

M.V. Iordache and P.J. Antsaklis. Synthesis of supervisors enforcing general linear constraints in Petri nets. *IEEE Trans. on Automatic Control*, 48(11):2036–2039, Nov 2003.

M. Jantzen. Language Theory of Petri Nets. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Advances in Petri Nets 1986*, volume 254 of *LNCS*, pages 397–412. Springer, 1991.

K. Jensen and G. Rozenberg, editors. *High-level Petri Nets*. Springer, 1991.

G. Jiroveanu and R.K. Boel. The diagnosability of Petri net models using minimal explanations. *IEEE Trans. on Automatic Control*, 55(7):1663–1668, July 2010.

R.M. Karp and R.E. Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147 – 195, 1969. ISSN 0022-0000.

M. Kloetzer, C. Mahulea, C. Belta, and M. Silva. An automated framework for formal verification of Timed Continuous Petri Nets. *IEEE Trans. on Industrial Informatics*, 6(3):460–471, 2010.

B.H. Krogh. Controlled Petri nets and maximally permissive feedback logic. In *Proceedings of 25th Annual Allerton Conference, University of Illinois, Urbana*, 1987.

K. Lautenbach and H. A. Schmid. Use of Petri nets for proving correctness of concurrent process systems. In *IFIP Congress*, pages 187–191, 1974.

D. Lefebvre. Estimation of the firing frequencies in discrete and continuous Petri nets models. *Int. Journal of Systems Science*, 32(11):1321–1332, 2001.

D. Lefebvre, E. Leclerq, N. El Akchioui, L. Khalij, and E. De Cursi. A geometric approach for the homothetic approximation of stochastic Petri nets. In *Int Workshop on Discrete Event Systems (WODES'10)*, pages 245–250, Berlin, 2010.

Z.W. Li and M.C. Zhou. Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems. *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 34(1): 38–51, Jan 2004.

J.L. Luo and K. Nonami. Approach for transforming linear constraints on Petri nets. *IEEE Trans. on Automatic Control*, 56(12):2751–2765, Dec 2011.

Z. Ma, Y. Tong, Z. Li, and A. Giua. Basis Marking Representation of Petri Net Reachability Spaces and its Application to the Reachability Problem. *IEEE Trans. on Automatic Control*, 2016.

Z.Y. Ma, Z.W. Li, and A. Giua. Design of optimal Petri net controllers for disjunctive generalized mutual exclusion constraints. *IEEE Trans. on Automatic Control*, 60(7): 1774–1785, July 2015.

C. Mahulea, L. Recalde, and M. Silva. Basic server semantics and performance monotonicity of continuous Petri nets. *Discrete Event Dynamic Systems*, 19(2):189–212, 2009.

C. Mahulea, C. Seatzu, M.P. Cabasino, and M. Silva. Fault diagnosis of discrete-event systems using continuous Petri nets. *IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 42(7):970–984, 2012.

G. Memmi and G. Roucairol. Linear Algebra in Net Theory. In Brauer (1980), pages 213–223.

P. Merlin. *A study of the Recoverability of Computer Systems*. PhD thesis, Univ. California, Irvine, 1974.

A.-L. Meyer. Discontinuity induced bifurcations in timed continuous Petri nets. In *Int. Workshop on Discrete Event Systems (WODES'12)*, pages 28–33, 2012.

M. Moalla, J. Sifakis, and M. Silva. A la recherche d'une métodologie de conception sure des automatismes logiques base sur l'utilisation des réseaux de Petri. In *Monographie AFCET: Sureté de Fonctionnement des Systèmes Informatiques*, pages 133–167. Eds. Hommes et Techniques, 1980.

J.O. Moody and P.J. Antsaklis. Petri net supervisors for des with uncontrollable and unobservable transitions. *IEEE Trans. on Automatic Control*, 45(3):462–476, Mar 2000.

T. Murata. Circuit Theoretic Analysis and Synthesis of Marked Graphs. *IEEE Trans. on Circuits and Systems*, 24(7):400–405, 1977.

J.H. Park and S.A. Reveliotis. Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings. *IEEE Trans. on Automatic Control*, 46(10):1572–1583, Oct 2001.

J. L. Peterson. Petri nets. *ACM Computing Surveys*, 9(3): 223–252, 1977.

C. A. Petri. *Kommunication mit Automaten*. PhD thesis, Technischen Hoschule Darmstadt, 1962.

C. A. Petri. *Communication with Automata*. Rome Air Development Center-TR-65-377, New York, 1966.

Petri Nets World. *Petri Nets Tool Database*, 2017. URL `https://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/db.html`.

J. Prock. A new technique for fault detection using Petri nets. *Automatica*, 27(2):239–245, 1991.

P.J. Ramadge and W.M. Wonham. The control of discrete event systems. *Procs. of the IEEE*, 77(1):81–98, Jan 1989.

C. Ramchandani. *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets*. PhD thesis, MIT, Boston, September 1973.

A. Ramirez-Trevino, I. Rivera-Rangel, and E. Lopez-Mellado. Observability of discrete event systems modeled by interpreted Petri nets. *IEEE Trans. on Robotics and Automation*, 19(4):557–565, Aug 2003.

L. Recalde, E. Teruel, and M. Silva. Autonomous Continuous P/T systems. In S. Donatelli and J. Kleijn, editors, *Application and Theory of Petri Nets 1999*, volume 1639 of *LNCS*, pages 107–126. Springer, 1999.

M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete-event systems. *IEEE Trans. on Automatic Control*, 40 (9):1555–1575, Sep 1995.

C. Seatzu, M. Silva, and J. Schuppen, editors. *Control of Discrete-Event Systems. Automata and Petri Net Perspectives*, volume 433 of *LNCIS*. Springer, 2013.

J. Sifakis. Uses of Petri Nets for Performance Evaluation. In *Measuring, Modelling, and Evaluating Computer Systems*, pages 75–93. North-Holland, 1977.

J. Sifakis. Structural Properties of Petri Nets. In J. Winkowski, editor, *Mathematical Foundations of Computer Science 1978*, pages 474–483. Springer, 1978.

M. Silva. Half a century after Carl Adam Petri's Ph.D. thesis: A perspective on the field. *Annual Reviews in Control*, 37:191–219, 2013.

M. Silva. Individuals, populations and fluid approximations: A Petri net based perspective. *Nonlinear Analysis: Hybrid Systems*, 22(11):72–97, 2016.

M. Silva and J. M. Colom. On the Computation of Structural Synchronic Invariants in P/T Nets. In G. Rozenberg, editor, *Advances in Petri Nets 1988*, volume 340 of *LNCS*, pages 387–417. Springer, 1988.

M. Silva and E. Teruel. A Systems Theory Perspective of Discrete Event Dynamic Systems: The Petri Net Paradigm. In P. Borne, J. C. Gentina, E. Craye, and S. El Khattabi, editors, *Symp. on Discrete Events and Manufacturing Systems. CESA '96 IMACS Multiconference*, pages 1–12, Lille, France, July 1996.

M. Silva, E. Teruel, and J. M. Colom. Linear Algebraic and Linear Programming Techniques for the Analysis of Place/Transition Net Systems. In G. Rozenberg and W. Reisig, editors, *Lectures in Petri Nets. I: Basic Models*, volume 1491 of *LNCS*, pages 309–373. Springer, 1998.

M. Silva, J. Júlvez, C. Mahulea, and C. R. Vázquez. On Fluidization of Discrete Event Models: Observation and Control of Continuous Petri nets. *Discrete Event Dynamic Systems: Theory and Application*, 21(4):427–497, December 2011.

R.S. Sreenivas. On the existence of supervisory policies that enforce liveness in discrete-event dynamic systems modeled by controlled Petri nets. *IEEE Trans. on Automatic Control*, 42(7):928–945, Jul 1997.

V.S. Sreenivas and M.A. Jafari. Fault detection and monitoring using time Petri nets. *IEEE Trans. Systems, Man and Cybernetics*, 23(4):1155–1162, 1993.

P. H. Starke. Reachability analysis of Petri nets using symmetries. *Systems Analysis Modelling Simulation*, 8 (4-5):293–303, 1991.

K. Trivedi and V. G. Kulkarni. FSPNs: Fluid Stochastic Petri Nets. In M. Ajmone Marsan, editor, *Application and Theory of Petri Nets 1993*, volume 691 of *LNCS*, pages 24–31. Springer, Berlin, 1993.

M. Uzam. An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions. *The International Journal of Advanced Manufacturing Technology*, 19(3): 192–208, 2002.

A. Valmari. Stubborn Sets for Reduced State Space Generation. In G. Rozenberg, editor, *Advances in Petri Nets 1990*, volume 483 of *LNCS*, pages 491–515. Springer, 1991.

B.F. Van Dongen, A.K.A. De Medeiros, and L.J Wen. Process mining: Overview and outlook of Petri net discovery algorithms. In *Trans. on Petri Nets and Other Models of Concurrency II*, pages 225–242. Springer, 2009.

C. R. Vázquez and M. Silva. Stochastic Continuous Petri Nets: An Approximation of Markovian Net Models. *IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 42(3):641–653, 2012.

C. R. Vázquez, A. Ramírez-Trevino, and M. Silva. Controllability of timed continuous Petri nets with uncontrollable transitions. *International Journal of Control*, 87(3):537–552, 2014.

S. Velilla and M. Silva. The spy: A mechanism for safe implementation of highly concurrent systems. *Annual Review in Automatic Programming*, 14:75–81, 1988.

G. Vidal-Naquet. Deterministic Petri net languages. In C. Girault and W. Reisig, editors, *Application and Theory of Petri Nets*, pages 108–202. Springer, 1982.

N. Viswanadham, Y. Narahari, and T.L. Johnson. Deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net models. *IEEE Trans. on Robotics and Automation*, 6(6):713–723, Dec 1990.

Y. Wardi, A. Giua, and C. Seatzu. IPA for continuous stochastic marked graphs. *Automatica*, 49(5):1204–1215, 2013.

Y.Q. Wu and C.N. Hadjicostis. Algebraic approaches for fault identification in discrete-event systems. *IEEE Trans. on Automatic Control*, 50(12):2048–2055, Dec 2005.