Petri nets and Automatic Control: A historical perspective

Alessandro Giua [a] Manuel Silva [b]

[a] *DIEE, University of Cagliari, Italy (giua@diee.unica.it)*

[b] *University of Zaragoza, Spain (silva@unizar.es)*

Abstract

The goal of this paper is to overview the historical development of the field of Petri nets (PNs) from a Systems Theory and Automatic Control perspective. It is intentionally not meant to be comprehensive: we limit ourselves to outline, through selected representative topics, some of the conceptual issues studied in the literature. In a first part we retrace the emergence of some basic net concepts to provide a broad view of the family of PN formalisms. Then we focus, more specifically, on the use of Petri nets within Automatic Control. Discrete net models have been considered since the middle of the 70s and starting since the late 80s have also been used for addressing classical problems, such as supervisory and deadlock control, state estimation, diagnosis, and so on. The double benefit is the ability to model a larger class of systems and to provide efficient algorithms for solving certain of those problems. We also discuss new approaches based on continuous and hybrid nets, which have been developed within the Automatic Control community.

Published as:

A. Giua, M. Silva, "Petri nets and Automatic Control: A historical perspective", *Annual Reviews in Control*, Vol. 45, No. 2, pp. 223–239, 2018. DOI:10.1016/j.arcontrol.2018.04.006

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

1 Preliminary overview

Born in a Computer Science milieu, as Carl Adam Petri was fond of saying, nets belong to the broad domain of *Systems Theory.* In the late fifties and at the beginning of the 60s of the past century, when the main focus was on local computations of mathematically intricate sequential problems, Petri developed a fresh approach to the theory of *concurrency* and *synchronization.* In fact, the title of his seminal work (1) is expressive: *Communication with Automata.*[1] Considering notions of *dependence* and *independence* of actions, *locality* of states and events were straightforwardly captured to support both *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 purely *autonomous* models, meaning by that *untimed* or, more precisely, possessing only a *qualitative* notion of time based on event ordering: earlier or later, possibly at the same time. Also they were *nondeterministic* models, a humble position leading to their logical study by considering all possible evolutions. The inception of *quantitative* time dates to the mid 70s, when issues related to performance evaluation, verification and control, such as throughput computation, optimal scheduling, etc., started to be considered. The works by (3), (4) and (5) are a few examples of representative early proposals for endowing PNs with a time structure. 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 above mentioned timed formalisms, one may associate input and output events with the firing of transitions to define *marking diagrams* (also *synchronized PNs*), which represent a clear generalization of Mealy machines in which the global state is replaced by a distributed one.

The above mentioned diversity of formalisms turns PNs into a theoretical framework or *paradigm* for the modeling of DES along their *life-cycle* (6), well suited 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, which naturally leads to state some necessary or sufficient conditions for its liveness as autonomous. Following the seminal work by (7), a broader perspective of so called *rank theorems* is provided by (8).

The first broad and organic perspective of works related to PNs is due to (9). It integrates the "structural" line deriving from Petri first proposal and the "automata-language" based approach,[2] together with *Vector Addition Systems* (10) and other graphical models for parallel computations, independently introduced in the USA since the late 60s. 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 in the series *Advances in Petri Nets*, published in the Lecture Notes in Computer Science (LNCS). Most of those contributions came from Computer Science.

Although with different degree of centrality, the family of formalisms known as Petri Nets have spread from Computer Science and Engineering (CSE) to other domains, including Automatic Control (AC) and Operations Research (OR) always supported by a solid background in Mathematics and Logic. We focus in this work mainly on the AC domain. Thus what is here presented is naturally a partial/biased view of the entire PN field. For a broader historical

---

[1] For its translation into English, (2).

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

perspective which traces the development of PN theory and applications in parallel with that of the PN community, see (11). The AC control community started, discovering PNs in the middle of the 70s. For example, (12), following the spirit of the times, use them for modeling, verification, analysis and implementation (hardwired, microprogrammed and programmed) of *logic controllers.*

Although the long period that has elapsed since 1962 has seen the appearance of an impressive number of contributions, 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 80s the quantitative notion of time 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 endless fight against the well-known *state-explosion problem* that affect DESs, new variants such as *continuous* or *fluid* and *hybrid* PNs, were introduced by the end of the 80s: this lead to a new 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 — in parallel with the rising interest of the AC community in DESs — 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. 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 a revised and enlarged version of (13), with additional discussions throughout and the inclusion of new sections on "early books" in the field and on about scheduling; it also contains an appendix providing a collection of bibliometric data about the development of the field. It 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 reviews some of the first books devoted to Petri nets, which have been instrumental in creating a sense of community. Section 4 deals with the use of PNs as dynamical models to address classical problems of AC, such as control, state estimation, diagnosis, scheduling, etc. Section 5 aims to sketch a bridge connecting control theory and engineering of continuous, hybrid and discrete event systems. Section 6 briefly mentions a few of the topics that we could not properly describe in depth in this paper. Finally a few promising areas that are open to future research are briefly discussed in Section 7, followed by the above mentioned Appendix.

2 Petri nets: from basic concepts to the modeling paradigm

Due to space limitations, just a few key steps in the long development of Petri net models are discussed in the sequel, starting with the seminal work of the field (1). In contrast with a widespread common vulgata, in the thesis of Petri 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 purely "strategic". He explicitly mentioned:

> I 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.

2.1 From Condition/Event to High-Level nets

The first net based formalism became what is known as Condition/Event nets, that are ordinary (i.e., arcs are unweighted) 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 60s, 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

Fig. 1. The University of Zaragoza granted Carl Adam Petri an honorary doctorate on April 15, 1999. The award was conferred during the celebrations the 25th anniversary of the foundation of the Engineering School (previously Centro Politécnico Superior). The picture was taken after the ceremony, on the central staircase of the Paraninfo building, and shows representatives of research teams from Australia, Canada, France, Italy, Spain and United Kingdom.

(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. 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, at will, both *top-down* and *bottom-up* approaches. Differently stated, models can be constructed not only by *refining* transitions or places, but also by *composing modules* through transitions (*synchronizations*, obtained by "synchronous product") or through places (*fusions*, obtained by "local state merging"); the advantage is that in any case the structure of modules is preserved.

During the first half of the 70s a second way of synchronization was introduced 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, (14) 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 (15),

> 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 (16) 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 and complementary ways of addressing analysis and synthesis problems in the PN framework.

An important logical extension of PNs was the introduction of *inhibitor arcs* (17), that allow the simulation of Turing Machines; soon this was followed by other similarly expressive extensions, such as *priority levels* on the firing of transitions. At the end of the 70s *High Level Petri Nets*, were introduced followed by more abstract (compact) formalisms, among which *Predicate/Transition Nets* and *Colored PNs* (9; 18). In high-level PN models tokens are *individualized* by means of labels (sometimes called *colors*). Information attached to 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 color domains are finite, colored PNs constitute only a "modeling convenience" with no greater expressive power. The interest of colored PNs for dealing with manufacturing systems was recognized since the early 80s. For example, in (19) part of a flexible workshop of the company Renault was modeled and validated. Colored nets have had an important impact in modeling industrial application case studies in quite different domains (see, http://cs.au.dk/cpnets/industrial-use/).

2.2 On properties and analysis techniques

Properties of PN models always depend on the net structure. They can be *behavioral*, if they 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, the existence of 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, in the second case most frequently the marking abstraction is expressed by the *existential quantifier*: for example, a PN is *structurally live* if *there exists* an initial marking such that the corresponding system is live.

During the 70s, 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* provide an exact or over-approximated 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 appeared in the late 80s and beginning of the 90s to reduce the size of the state-space to be searched by a *model checking* algorithm, are the *stubborn sets* (a partial order technique) (20) and the identification of *symmetries* (21). Very recently in (22) 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. (23; 24). As a proof of the strong "coherence" among the different abstraction levels in the net family of formalisms, during this decade and the 90s the above techniques were extended to Colored PNs (25).

The complexity of Petri net decision procedures and the properties of PNs as *language generators* have been studied since the early 70s (26). 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 (27). 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 (28).

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 of interest. If the transformation has polynomial complexity, but the analysis of the transformed system is exponentially "cheaper", the advantage is obvious. For example, (29) showed that the analysis of properties such as *boundedness*, *liveness* or the existence of *home states* can be computationally simplified 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 (8). However, the existence of irreducible net systems for simple properties shows that the method, even if very interesting in practice, 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 (30). From a purely AC perspective, (31) 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 (32) and (33); 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 (34). 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 (35) or can increase the (generalized) 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.

As a final comment, it should be stressed that — especially in the case of bounded Petri nets — whenever more efficient structural techniques are not applicable to solve a particular problem, one can always resort to enumerative techniques. By constructing the reachability or coverability graph, automata based analysis techniques can thus be applied.

2.3 From timed interpretations to the Petri net modeling paradigm

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 probability distribution functions, 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 chain 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, which include structural decomposition, tensor algebra methods, symmetries, etc. (36; 37). This subfield, which is still very active, started in the late 70s and reached its maturity by the end of the 90s (38; 39). Timed models are also used in real-time applications looking for correctness, from logical properties such as deadlock-freeness, to explicit response time-bounds. An interesting approach for dependability analysis based on timed stochastic nets was proposed by (40). They describe a methodology to construct dependability models using generalized stochastic Petri nets and also a more powerful model called *stochastic reward nets*.

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 for non-Markovian models. A key approach to *Discrete Event Simulation* was proposed in 1976 by (41), where a "model-driven" perspective was introduced, leading to a separation of the model construction from the simulation techniques. [3] 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-70s, in the area of *Programmable Logic Controllers* 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 (see in the next section the comments on the book by (43)).

Petri Nets at different *levels of abstraction* (C/E nets; Predicate/Transition nets; Colored PNs; Object Oriented PNs, etc.) and possibly endowed with a large number of *interpreted extensions* (Marking Diagrams; Batches PNs; deterministic, stochastic or possibilistic-fuzzy timing structures, etc.) represent 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, as shown in Fig. 2 PNs constitute a broad *modeling paradigm*, constructed as a ”Cartesian product” of different levels of abstractions and of different extensions by interpretation (6). Different formalisms may be used in the study of a given system along its *life-cycle* (for example, in the design and analysis of the manufacturing cell considered in (11)).

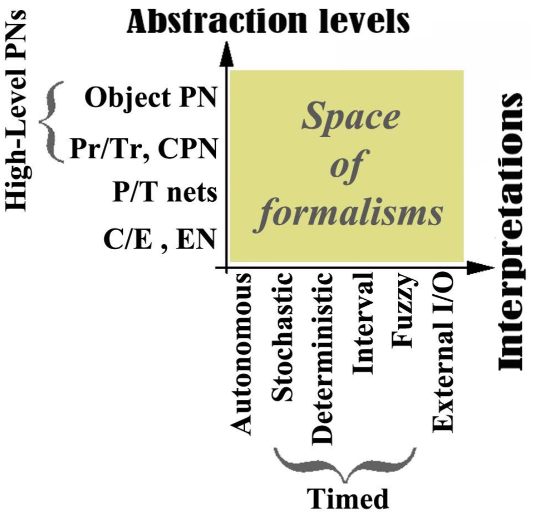

Fig. 2. A possible view of the PN-based modeling paradigm as a ”Cartesian product” of different levels of abstractions and of different extensions by interpretation.

As a final comment, let us mention that a rich diversity of tools can be found on the (44) repository for the simulation and analysis of different Petri net models.

3 Some ”early” books on Petri Nets theory and Applications

The first books on Petri net theory and applications appeared in the 80s.

collects the material presented during an advanced course that was held the previous year in Hamburg. It contains a significant number of the results available by the end of the 70s, integrating the “structural” or “systems theory” approaches (based on the use of invariants, etc.) closer to operations research — more properly inspired by Petri — and the “automata” approaches. It also presents other graphical models for parallel computations, independently introduced in the USA in the late 1960s. An interesting feature, motivated by the novelty of the field, is a dictionary of basic notions of Net Theory that aims to clarify the problems posed by the new terminology.

---

[3] The formalism *Discrete Event System Specification* (DEVS) was initially based on an extension of Moore’s automata, but subsequent extensions have been developed. For a recent historical perspective on DEVS, see (42).

A basic and systematic perspective on modeling and analysis with Place/Transitions is due to (45). This book is written in a very intuitive and readable style. Basic analysis tools such as the reachability graph and matrix equation are presented. Different classes of PN languages and complexity and decidability issues are also explored. Although at the time the book was written it had not yet been proved that the marking reachability problem is decidable, the constructions to reduce many other problems to reachability are still interesting today. A few extended formalisms (beyond P/T nets) are also presented.

The first book in German is due to (46). It is also mainly focused on Place/Transition nets, but in addition mentions timed models and high-level nets. It contains an interesting collection of results related to: concurrency and conflicts; reachability; liveness, equivalent markings and persistent transitions; reduction and invariants; fairness and synchrony, structural properties, siphons and traps. The book also contains a chapter on software tools. The monograph due to (47), also written in German, was later translated into English (48). Characterized by a more axiomatic style, it is structured into three main parts. A first part deals with Condition/Event-Systems and includes a chapter devoted to synchronic distances. Later Place/Transition nets are studied and their analysis is carried out, centered on net invariants. Finally nets with individual tokens, examples of high-level nets, are presented. In an appendix, a bibliography covering the development of the field during its early years is provided.

The first book in French is (49). This is a collective pseudonym chosen by a French group of researchers composed by Ch. André, G. Berthelot, C. Giraud, G. Memmi, G. Roucairol, J. Sifakis and G. Vidal-Naquet. This work is mainly centered on Place/Transition nets, and structured into two volumes. The first one presents a rich collection of key results concerning analysis techniques, including coverability tree, linear algebra and reduction. The second volume is devoted to modeling: it presents several additional structures such as abbreviations (high-level nets), extensions (nets with inhibitor arcs), and non-autonomous nets (timed, in particular). Many modeling examples and a few implementation issues are also discussed.

Actually written in 1982, the first book in Spanish is due to (50) and is targeted to both the Automatic Control and Computer Engineering communities. It is structured into two parts. The first one is devoted to: modeling of net systems, including top-down and bottom-up approaches; net transformation rules, such as model simplification; analysis techniques for autonomous net models, including reachability, reduction, and structural approaches. Considerations on the analysis of non-autonomous net systems are also provided. The second part is devoted to different kinds of implementation technologies: hardwired, microprogrammed, Programmable Logic Controlled and microcomputer based. Each of these topics is comprehensively treated in a separate chapter.

(43) present an introduction to PNs that was particularly targeted to Control Engineers. They start from Grafcet, a language for the specification and implementation of logic controllers, and enlarging this modeling setting they introduce Petri nets. It should be noted, as a curiosity, that this progression takes the opposite direction with respect to the development of the French standard Grafcet, which was actually inspired by Petri nets. Later Grafcet was redefined as the Sequential Function Chart (SFC), a graphical language for programmable logic controllers and adopted by the International Electro-technical Commission (included in IEC 1131-3 standard).

4 Petri nets as discrete event models for control systems

In this section, we review the development of PN research within the area of DES in Automatic Control. It shows how they have been used to address classical problems of control systems in a broad sense, including analysis, control, diagnosis, state estimation and observability, identification, etc. A similar analysis concerning fluid PN models can be found in subsection 5.2.

DES have been formally considered in the framework of AC since the 70s, being representative of such tradition the two *International Symposia on Discrete Systems* sponsored by IFAC in Riga (1974) and in Dresden (1977) which focused on the theory and application of switching network theory and of applied automata theory. In particular, in the 1977 meeting seven papers dealt with Petri nets, all of them from European authors.

In the United States, the DES community in AC originated in the 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 (51). Also a series of informal workshops (such as the *NASA Workshop on Discrete Event Systems and Artificial Intelligence*, organized in Princeton in 1990) were instrumental in the development of a DES community in the USA and several results based on PNs were presented in these venues.

Fig. 3. Some early books on Petri nets. From left to right and top to bottom: Brauer (1980); Peterson (1981); Reisig (1982); Brams (1983); Silva (1985); David and Alla (1989).

In Europe the DES community has centered around the WODES series (Workshop on Discrete Event Systems) whose first event was held in Prague in 1992. In the published proceeding of WODES'92 (52) 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.

4.1 Supervisory control

Supervisory control is a fundamental theory for the control of DESs that has been proposed in the 80s by (53). The reader may also refer to a companion history paper by (54) on the history of Supervisory Control which is included in the same special issue of Annual Reviews in Control.

Supervisory control is very general and model independent approach. The original contributions and most of the subsequent developments in this domain focused on automata, that are intuitive models useful for presenting basic concepts. However, it is well known that automata have a limited modeling power with respect to Petri nets: they can only describe finite state systems and lack explicit primitives to model important behavioral features such as concurrency and synchronization. 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 (55).

4.1.1 Language specifications

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 considered within this framework. A first example was due to (51), who used Petri nets to show that when events may occur concurrently 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 (56). 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 (57; 58). The existence of infinite state supervisors has also been discussed by (59).

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 (60).

Unfortunately, due to the presence of uncontrollable transitions, one needs to refine this structure to avoid reaching undesirable — e.g., uncontrollable or blocking — marking. The set of undesirable markings usually does not have a special structure and it is not obvious how the net system can be refined to prevent reaching them. A general approach that can be used to refine a net model is based on the *theory of regions* (61): 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.

Most of supervisory control approaches based on automata focus on the *non-blocking* property, which requires that from any reachable state a marked (or final) state should be reachable. A similar interesting property, which is called *liveness* and is related to the occurrence of events, can be defined in the framework of PNs. This property requires that from all reachable state any transition firing can *eventually* occur. Liveness enforcing by supervisory control has been explored in several works and is considered a complex problem especially when some of the transitions of the net are not controllable. (62) showed that the existence of liveness enforcing control policy is in general undecidable but decidable for bounded nets. In the case of bounded nets (63) presented an efficient approach that uses a PN technique called *unfolding* to derive a liveness enforcing supervisor. However it has been been later pointed out by (64) that the approach is only applicable to a restricted class of nets.

In conclusions, 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.

4.1.2 State specifications

Another very active subdomain of supervisory control deals with the control objective of preventing a plant from reaching a set of undesirable states. This is a problem that can be addressed leveraging the many features that directly pertain to 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 the subset of its behavior that is mostly relevant to solve a given control problem.

A first approach was presented by (65) considering a controlled Petri nets model, i.e., a net system endowed with external inputs that can enable or disable the firing of a subset of controllable transitions. They showed that it is possible to efficiently design control synthesis algorithm to preventing subset of places from getting marked by the analysis of the uncontrollable paths that lead from a given controllable transition to this set. This approach has been shown to be particularly useful to solve control problems that arise in automated manufacturing systems (66).

State specifications were also considered by (67): it should be noted that these authors adopted the peculiar choice of presenting their results in the framework of Vector Addition Systems, a model that is equivalent to PNs although less intuitive and less familiar to the control community.

A large number of works deal with the control synthesis for a wider class of state specifications called *Generalized Mutual Exclusion Constraints* (GMECs), which consider set of legal markings defined as the integer solutions of a convex set. These constraints were introduced by (68) and also proposed by (69). 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 exploring 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.*

The major drawbacks of the approach based on GMECs, as pointed out in (68; 70), is the fact that when the net contains uncontrollable or unobservable transitions a monitor based solution may not be feasible, in the sense that it may require disabling uncontrollable transitions or detecting the firing of unobservable transitions. We say in this case that the monitor, and the corresponding GMEC, is uncontrollable or unobservable. The original solution proposed by (70) to overcome this problem consists of an elegant algebraic procedure to compute a more restrictive — but controllable and observable — GMEC that can be applied to general nets. This allows one 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 inspired by this approach followed. The extension to constraints also involving firing vectors was presented by (71). Assuming actuators and sensors associated to monitor have a cost, (72) modified the Moody and Antsaklis' procedure to determine sub-optimal monitors of minimal cost. (73) considered the problem of reducing the number of monitors on safe net to lower the cost of implementing a controller. (74) proposed constraint transformation rules that ensure maximal permissiveness for subclasses of nets. (75) proposed a branch-and-bound approach to derive a minimal supervisor guaranteeing the attainment of an arbitrary set of static and behavioral specifications in a maximally permissive way.

Note that the research in this area of supervisory control is still active. We mention a recent work by (76) where non-convex legal marking sets described by disjunctions of GMECs can be enforced on nets where all transitions are controllable by a control structure called monitor-switcher.

4.2 Deadlock analysis and control

Deadlock freeness is a basic property that has, 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. For this reason deadlock freeness is a property that must be always ensured and which is particularly relevant in many automation problems where appropriate strategies should be adopted to enforce it.

Deadlock analysis and control for PNs is based on structural approaches and 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) (77). For some net subclasses, such as Free-Choice nets, DF implies liveness. Nevertheless, the siphon/trap condition is computationally hard to check. (78) formulate a deadlock detection method by solving integer linear programming such that either a complete siphon or state enumeration is not necessary.

An alternative to the siphon/trap based analysis are 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 leads to a necessary and sufficient condition for liveness that can be tested in polynomial time (checking for consistency, conservativeness and a bound on the rank of the incidence matrix). 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 (8).

What to do if a given PN deadlocks? 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* (79). 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 (80): 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. A generalization of this approach to a other class of nets was presented by (81). The case of nets with uncontrollable and unobservable transition was discussed by (82).

For more general classes of Petri nets modeling automated manufacturing systems, a deadlock avoidance algorithm with polynomial complexity was developed by (83). Efficient solutions have also been proposed by (84), using the concept of elementary siphons whose number is *linearly* bounded by the structural size of a Petri net.

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 (85). As a further improvement of this approach, (86) 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.

4.3 Scheduling

To schedule is to define a plan of procedure in order to achieve a goal (or set of goals, in a multi-criteria case). Since the 60s, much effort was devoted to this topic by the Operations Research (OR) community.

This kind of problems is important not only in Operations Research, but also in Automatic Control, having the basic ingredients of any optimization problem: one or several (multi-criteria) objective function and some dynamic constraints. Scheduling is thus a control problem in the DES framework.

In a scheduling problem, time is usually explicitly represented in the dynamic model (e.g., Time Petri Nets (4)), and the goal is frequently expressed as a cost minimization or a befit maximization. For example, in the acyclic case it is frequently required to minimize the *makespan*, i.e., the time required to reach a target marking from a given one. This performance index represents the total time that elapses from the beginning to the end of the task to be performed. In a cyclic case — periodic schedules abound in practice in domains like air flight, train or manufacturing control — it is common to minimize the cycle-time, i.e., the time that the system requires to go back to the initial marking after performing a set of actions. In bounded Petri Net systems, a cyclic scheduling corresponds to the execution of a T-semiflow — unfortunately non necessarily minimal! — which in turn leads to the notion of $k-$periodicity. *Strict* periodic schedules (i.e., corresponding to $k = 1$) are not optimal in general, as illustrated in Fig. 4.(1), but they can be easier to compute and implement.

PN scheduling aims to define a timed (partial) ordering on the firing of transitions. When the Petri Net model interleave cooperation and competition relationships (because conflicts and synchronizations are simultaneously present), scheduling implement conflict resolution strategies, but not only. Paradoxically, just delaying actions may improve a cost function as shown in Fig. 4.(2):

One of the first works where Petri nets were proposed as a suitable model for scheduling was probably the paper by (88). He surveyed different formalisms for the modeling and analysis of parallel computing (the term used at that time was multiprocessing) and how to schedule the assignment of available resources (i.e., means to perform the activities) to parallel processes. The paper only considered logical models because timed Petri nets had not yet been defined at that time. What was considered the most interesting feature of Petri nets in this context was the explicit notion of concurrency and the possibility to easily describe counters. Among the early works about the scheduling of timed PNs we mention a paper by (89), where the practical modeling power of the PN formalism is used to represent functional precedence and temporal relations among the operations, and resource constraints (removable, as machines, or tools; or not removable, as energy). The main advantage is that, interleaving *functional* and *resources* constraints, it extends the kind of representable problems. For example, systems more complicated than job-shops and flow shops can be modeled.

It should be pointed out that, according to the time interpretation, the firing of transitions can occur under *single-server semantics* (i.e., no self-concurrency) or *multiple-server semantics*: this has important consequences on the

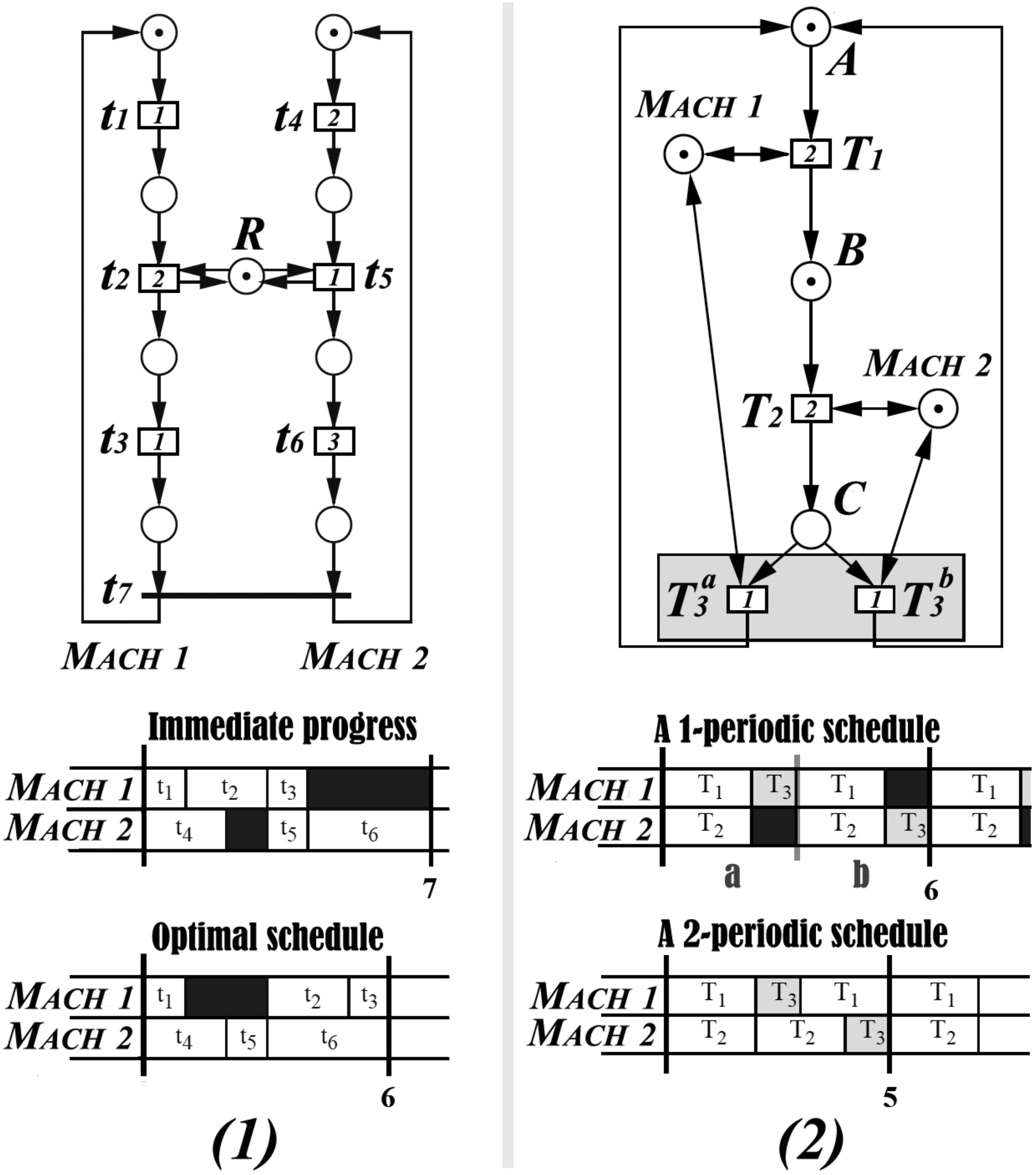

Fig. 4. Some remarks on optimal schedules (87). Let us consider the two deterministically T-timed net systems in figure (time durations are specified inside the boxes representing the transitions). (1) *Immediate progress* versus *optimal schedule:* t_2 could acquire resource R before t_5, but delaying its firing leads to a faster cyclic behavior (i.e., a local delay leads to a faster global behavior). (2) *On k-periodic schedules:* the firing of the T-semiflows $(1,1,1,0)^T$ or $(1,1,0,1)^T$ in isolation takes 3 time units and correspond to a strictly periodic schedule with $k = 1$ (the initial marking is reproduced every 3 units). The combined firing of both T-semiflows takes 5 time units, but fires the same transitions that the previous schedule fired in two periods: so the cycle-time is only 2.5 time units and the initial marking is reproduced in each period at time 3 and 5. Obviously, this duration cannot be reduced anymore, i.e., the optimal schedule is $k-$periodic with $k = 2$.

computational approaches. A more technical problem was addressed by (90), which study some of the properties of deterministic-timed schedules in PNs under single-server semantics. It has been proved that the set of schedules issued from a firable sequence of the underlying net model has a minimum element called *earliest schedule* of the sequence. For deterministic timed bounded net systems, optimal schedules are $k-$periodic. Optimal solutions correspond to *critical circuits* of the so called *earliest state graph*, which unfortunately is frequently too large. In particular, the computation of earliest schedules is usually an NP-problem, even for restricted net subclasses such as live and bounded free-choice systems (91).

In scheduling, *control points* are steps where decisions over the future evolution should be taken: examples include resolving a conflict or delaying the firing of a transition. This OR centered terminology was later reformulated in Automatic Control — under a broader perspective — by the Supervisory Control theory, that classifies transitions into *controllable* (i.e, once enabled, their firing can be delayed possibly for an arbitrarily long time) and *uncontrollable*: this notion applies to both timed and untimed net systems.

Most frequently, the computation of "good" schedules is computationally very expensive (NP-hard). In many cases rather than an optimal or suboptimal solution, one has to settle for a feasible one. Therefore, approaches that can reduce/simplify the computational effort are of paramount importance. Two classical classes of approaches of this type are the following.

(1) *Heuristic rule-based systems.* Among earliest works on the field, (92) used high-level Petri nets for modeling, analysis, and software implementation of controllers for flexible manufacturing systems. In particular a design procedure for a real-time scheduler as an expert system is proposed. While the considered model is a purely logical Petri net, timing information can be incorporated in the expert system.
(2) *Graph search algorithm.* They may use dynamic programming strategies, branch and bound approaches, A*, etc. (93) showed how PNs combined with heuristic search (based on A* graph search) could provide a new scheduling method for flexible manufacturing systems. Considering a timed Petri net model, the idea was to generate and search a partial reachability graph to find an optimal or near optimal feasible schedules. Different heuristic functions for efficient search were considered. The most interesting peculiarity of Petri nets in this context was their modeling power, because they can handle features such as routing flexibility, shared resources, lot sizes and concurrency.

Moreover, mathematical programming, net-transformation (reduction) and meta-heuristics (e.g., Genetic Algorithms, Simulated Annealing, Tabu Search) based approaches are used. An overview of the literature appeared during the years 1989-2005 in production and operations management journals, is provided by (94). Some complementary references can found in chapters 5-7 in (95).

In the 1990s as the semiconductor manufacturing industry was booming and was a major drive in the economic growth of many Asian countries, there was a great interest in the use of scheduling techniques to optimize the fabrication of these devices. A survey on the use of Petri nets for the analysis and control of semiconductor manufacturing systems was presented by (96). This work considered both logical and timed Petri net models and showed how different net formalisms could be used successfully to study different stages of these complex systems and to solve different problems. The focus was on those aspects of a manufacturing process that are peculiar to the semiconductor industry, including the scheduling problems that are particularly relevant in this context.

(97) was probably the first researcher to specifically model with Petri nets *cluster tool* processing. This was motivated by the growing importance of this manufacturing technology at the end of last century as the semiconductor industry was migrating to larger wafer sizes, and smaller device geometries. The steady-state analysis of cyclic scheduling processes are proposed. Subsequent works also dealt with this class of semiconductor manufacturing systems.

(98) consider models for cluster tool with a reentrant wafer flow. This complicates scheduling and control of the cluster tool and often causes deadlocks. They derive a necessary and sufficient condition for preventing a deadlock. They also show that the cycle time for the asymmetric choice Petri net model for a reentrant wafer flow can be easily computed from the equivalent event graph model based on mixed integer programming.

An additional improvement in this particular domain is due to (99). The authors consider the scheduling of cluster tools with residency time constraints, that are commonly encountered in some wafer fabrication processes (e.g., low pressure chemical vapor deposition). A particular Petri net model is developed for the system and used to address both the schedulability and scheduling problems. In particular necessary and sufficient conditions are given for the schedulability of a single-arm cluster tool with residency time constraints. In addition, a closed form scheduling algorithm is developed to find an optimal periodic schedule, if it exists.

It is impossible to overview the many technical approaches to scheduling in a brief perspective as this one. Anyhow, somehow surprisingly, the number of contributions to the subfield is relatively "too small" compared with other control subtopics, as deadlock-freeness control.

4.4 State estimation, diagnosis and identification

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

4.4.1 State estimation

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 [4]. The agent that can reconstruct the value of the state is called *observer.* In the case of PNs one is interested in reconstructing the current marking but also possibly the firing sequence that has produced the given observation.

In PNs one usually assumes that measurable outputs are labels assigned to transitions, which are observed when they fire. This model has been originally introduced to define the family of Petri net languages of which there are several classes depending on the labeling function and on the set of final states (27). In the discrete event literature, such a class of models is called *partially observable Petri nets* (POPNs): here nondeterminism may occur due to the empty labeling of some transitions (called *unobservable*) or to a non-injective labeling (transitions sharing the same label are called *undistinguishable*). A different model where also the token content of some *observable* places can be measured was presented by (100), but the this class of models was later shown to be language equivalent to the class of POPNs by (101). Recently, new classes of observation structures that extend the modeling power of POPNs where presented by (102) and the relationship between then was explored.

While the vast majority of the literature on state estimation using PNs is based on POPNs, it is interesting for historical reasons to mention that one of the first approaches in this area, presented by (103), considered a different setting, more similar to the state estimation of time-driven systems. 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. In this same setting, (104) explored state feedback control with observers and proposed to use time-out information (bounds on the maximal time required by a transition firing) to improve the marking estimate to recover from controller induced deadlocks.

Also, the previously mentioned paper by (100) considered *interpreted* PNs, i.e., nets with both inputs and outputs — including observations of transitions labels and of the marking in subset of places — and studied the observability property of this class, i.e., the possibility of exactly reconstructing the marking value at each step.

When POPNs are considered, due to their intrinsic non-determinism the observer cannot always reconstruct the exact value of the marking at each step, but aims to reconstruct the set of *consistent markings*, i.e., the set of markings in which the system could be given the collected observation. A first approach of this type was presented by (105) but has now a purely historical interest, since it was applicable to a very restricted class of nets, both in term of labeling function (the only cause of non-determinism was due to undistinguishable transitions) and of net structure (the non-deterministic transitions could not share input/output places). A more interesting approach, which was later applied and extended in different settings (diagnosis, opacity, etc.) and contained *in nuce* the idea that was later developed into the analysis technique called *basis marking approach*, was presented by the same authors in (106).

In the last ten years many new original contributions in this active area of research have been presented. (107) considered a problem of practical relevance, namely designing, by a suitable choice of sensors, an observation structure that ensures the observability (exact estimation of the marking at each step) of a given POPN for arbitrary initial markings. Concerning the estimation of firing sequences, (108) proposed an efficient approach to determine an optimal (least-cost) transition sequence that can explain (or produce) a given observation. The same authors also addressed the problem of estimating the initial marking (assumed unknown) of a POPN, as opposed to estimating its current marking (109).

More recent results concern the problem of marking estimation in a non-logical setting. An approach for probabilistic estimation was presented by (110). (111) consider the problem of estimating, using linear algebraic approaches, the marking of timed nets whose structure is backward/forward conflict-free. (112) address the marking estimation for time nets constructing a Modified State Class Graph.

---

[4] In the case of controlled systems the knowledge of control inputs is also assumed.

4.4.2 Fault diagnosis

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* (113). Other approaches were based on monitoring the tokens in P-invariants (114) or backfiring transitions to determine if a given state is invalid (115). Years later, (116) 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 (117) 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 events 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 (118), an approach based on *event estimation* (119), the use *minimal explanations* (120) and *basis markings* (121) to avoid the full construction of the reachability set, or on-line approaches based on integer programming (122; 123). PNs have also been used to extend this approach to infinite state systems (124) and to times systems (112; 125). 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 (126): this a topic of great relevance in the context of multi-agent systems, internet of things, etc.

The reader may also refer to a companion history paper by (**?**) on the history of Diagnosis and Opacity which is included in the same special issue of Annual Reviews in Control.

4.4.3 Identification

This classical problem consists in constructing a mathematical model of a dynamical system starting from measured data. In the domain of PNs, identification is also known with the name of *net synthesis* (127) and is closely related to the process mining technique called *discovery*, i.e., constructing an unknown model based on an event log (128).

Process mining approaches usually can handle large amounts 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 (129). 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 (130) or on-line (131). Finally the identification of fault models has also been addressed (132; 133).

5 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) (134). Fluid or continuous PNs are obtained by a simple relaxation. The underlying idea is not really new — consider, for instance, the Lotka-Volterra equations that appeared in 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 a paper presented by (135) at the 8th European Workshop on Application and Theory of Petri Nets in Zaragoza. As explicitly stated by (136), the source of inspiration was the fluidization of models for the performance evaluation of production lines (manufacturing domain). At the same workshop in Zaragoza was also presented a paper proposing the systematic use of linear programming techniques for the structural analysis of PNs based on their *fundamental* or *state-transition equation*; a revised version of this work is (34). 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 fluid 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. (137) provides a recent tutorial perspective on the fluidization of PNs.

5.1 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, also if 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 as the emptying of traps (138). 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 (139).

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 synchronization 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 (136), while many results concerning ISS-TCPN are summarized in (134). 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 (140). 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 timed 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 (141). 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. Complex behaviors may appear even in restricted classes of nets such as fluid Mono-T-Semiflow systems (see Fig. 5).

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*.

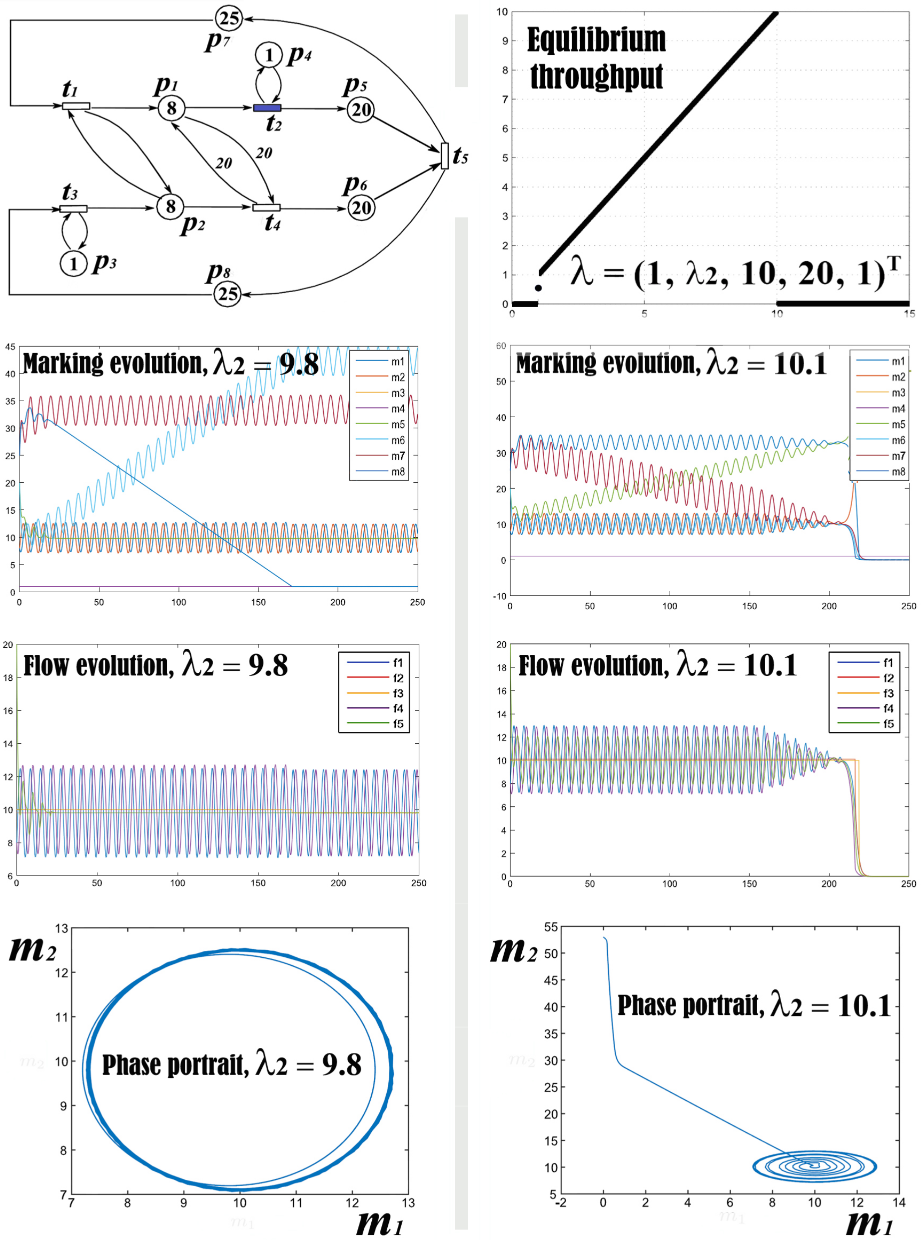

Fig. 5. A fluid Mono-T-Semiflow net system. Under infinite server semantics, it may exhibit bifurcations, discontinuities, and different kinds of oscillations (137; 142): (1) The net model. (2) When λ_2 is changed, the equilibrium throughput has two discontinuous jumps (the second is non-monotonic; i.e., increasing the firing speed the system performm slowly); therefore three regions appears (in the first and the last, deadlocks are reached at different markings; in the central one, it grows linearly, while the system oscillate). (3) Marking and flow evolution, and phase portrait (m_2 vs. m_1) with two different but close values for λ_2. In one case, an oscillatory behavior of the marking (and hence of the throughput) may remain forever; in the second one, a final trashing phenomena appears.

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* (134), 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 (137): (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 (143). Stochastic approximations (144; 145) 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 (146).

5.2 On the use of fluid models

Since the early works it was clear that fluid PNs enjoy important advantages (138). 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* 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 (134). (147) 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 (148). 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 fluid systems, *mathematical programming* methods can be applied, for example to compute: an optimal initial marking, an optimal routing rate at conflicts, or an optimal steady-state (149). See also chapter 18 in (150). In (151) 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 (152). 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, (134), and chapter 20 in (150)). Among problems that did not receive yet satisfactory solution is the decentralized control of ISS-TCPN with uncontrollable transitions, a problem of relevance for "populated" systems of large dimension.

5.3 A brief perspective on hybrid PNs

The extension from (discrete) net models to *Hybrid PNs* (HPNs) did follow many complementary lines.

HPNs with continuous places. The first type of models to be defined consisted of two types of places: discrete places, containing tokens; continuous places, containing fluid. Thus, the marking of discrete (resp., continuous) places represents the discrete (resp., continuous) component of the hybrid state.

The continuous dynamics are usually represented by relaxing the firing of a *subset* of transitions, as in the continuous nets of David and Alla: such is the case of the first work on HPNs presented by (153). For this model, called *hybrid Petri nets* tout-court, the authors develop analysis techniques under single server semantics based on the extension of the classical reachability graph. Models of this type usually derive from the relaxation of deterministic timed PNs. Nevertheless, untimed models and analysis techniques are also of interest.

A different model, presented by (154), assumes that the marking of continuous place is produced by *fluid arcs* and derives from the relaxation of generalized stochastic Petri nets (38). The evolution of this model can be studied by means of a system of coupled differential equations which, in the special case of nets with a single continuous place, reduce to a simpler differential equation.

HPNs with continuous variables. An alternative basic approach to define HPN formalisms, is inspired by *hybrid automata* (155). In this class of models, a discrete PN is used to describe the event-driven dynamics and the discrete part of the hybrid state is represented by the net marking. The continuous part of the state is described by additional variables, whose dynamics is ruled by differential algebraic equations (DAE) not represented in the net structure. The first examples of this class are the DAE-nets of (156) and of (157).

Other models. (158) defined *Differential PNs* allowing the marking of a place to also take negative real values so as to describe non-positive dynamics. Further extensions, based on the concept of high-level Petri nets, have also been defined: by associating colors to the transition firings and to the markings, (159) showed that arbitrarily complex continuous dynamics can be modeled. However, all these models did not have much success due to the fact that they are quite difficult to analyse and do not keep many of the peculiar structural features of PNs.

All the hybrid PN formalisms mentioned above appeared in the 90s and by the end of that decade, in the broader framework of hybrid systems, several of them were reviewed by (160). A special issue of the journal Discrete Event Dynamic Systems, devoted to hybrid PNs by (161), also provides an accurate description of the state of the art at that time.

New models were also defined in the new century. (162) defined *First-Order Hybrid PNs* where the firing of continuous transitions is driven by control inputs (as opposed to describing autonomous evolutions) and derived a framework for the analysis and control of these models based on linear and integer programming, extending to a hybrid setting PN structural theory. *Batches PNs* were defined by (163) to describe high-throughput systems where there is a transfer of material that moves in space with preassigned speed and can change its density when accumulation occurs; the stationary behavior of this model can also be characterized using structural analysis as shown by (164). Markovian Hybrid Petri nets were introduced by (165). Provinding a "legitimization" for HPNs with continuous places, they approximate Markovian Petri nets to overcome the state-space explosion that characterizes the latter discrete model.

Finally, we mention that while many HPNs are a complex model, not always amenable to formal analysis, due to their generality they are convenient for modeling and simulation of a large classes of systems of practical interest. Examples can be found in automated manufacturing (166) and in particular semiconductor manufacturing (167) and assembly lines (168). Other examples include batch processes (169), urban traffic (170) and supply chains (171).

6 Other topics

In this historical perspective we have tried to evaluate the impact of Petri nets in Automatic Control. While many topics have been briefly reviewed, we had to pass over some others.

In particular, we abandoned the idea of analyzing in some detail the literature on timed Petri nets models, which are mostly used to address issues of performance analysis and optimization, restricting ourself to point out in Section 2 some of the main historical landmarks in the development of this area of research. We mention, however, that a companion history paper by (172) included in the same special issue of Annual Reviews in Control, presents the history of max-plus algebra which has been used for the analysis of an interesting subclass of Petri nets, *deterministic timed Marked Graphs.*

Issues related to the implementation of PN based logic controller have not been considered in this paper. In addition, we did not discuss the many applications of Petri nets, such as manufacturing, logistic, health-care systems, traffic control, biological systems, communication protocols. For a brief consideration of these two topics the reader may consider the sections *Logic controllers and their implementation* and *On applications and maturity* in the previously mentioned paper by (11).

7 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 (173), 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

[1] C. Petri, Kommunication mit Automaten, Ph.D. thesis, Technischen Hoschule Darmstadt (1962).
[2] C. Petri, Communication with Automata, Rome Air Development Center-TR-65-377, New York, 1966 (1966).
[3] C. Ramchandani, Analysis of Asynchronous Concurrent Systems by Timed Petri Nets, Ph.D. thesis, MIT, Boston (September 1973).
[4] P. Merlin, A study of the Recoverability of Computer Systems, Ph.D. thesis, Univ. California, Irvine (1974).
[5] J. Sifakis, Uses of Petri Nets for Performance Evaluation, in: Measuring, Modelling, and Evaluating Computer Systems, North-Holland, 1977, pp. 75–93 (1977).
[6] M. Silva, E. Teruel, A Systems Theory Perspective of Discrete Event Dynamic Systems: The Petri Net Paradigm, in: Symp. on Discrete Events and Manufacturing Systems. CESA '96 IMACS Multiconference, Lille, 1996, pp. 1–12 (July 1996).
[7] J. Campos, G. Chiola, M. Silva, Properties and Performance Bounds for Closed Free Choice Synchronized Monoclass Queueing Networks, IEEE Trans. on Automatic Control 36 (12) (1991) 1368–1382 (1991).
[8] M. Silva, E. Teruel, J. M. Colom, Linear Algebraic and Linear Programming Techniques for the Analysis of Place/Transition Net Systems, in: G. Rozenberg, W. Reisig (Eds.), Lectures in Petri Nets. I: Basic Models, Vol. 1491 of LNCS, Springer, 1998, pp. 309–373 (1998).
[9] W. Brauer (Ed.), Net Theory and Applications, Vol. 84 of LNCS, Springer, 1980 (1980).

[10] R. Karp, R. Miller, Parallel program schemata, Journal of Computer and System Sciences 3 (2) (1969) 147 – 195 (1969).
[11] M. Silva, Half a century after Carl Adam Petri's Ph.D. thesis: A perspective on the field, Annual Reviews in Control 37 (2013) 191–219 (2013).
[12] M. Moalla, J. Sifakis, M. Silva, A la recherche d'une métodologie de conception sure des automatismes logiques basÃ©e sur l'utilisation des réseaux de Petri, in: Monographie AFCET: Sureté de Fonctionnement des Systèmes Informatiques, Eds. Hommes et Techniques, 1980, pp. 133–167 (1980).
[13] A. Giua, M. Silva, Modeling, analysis and control of discrete event systems: a petri net perspective, in: Proceedings of 20th IFAC World Congress, Vol. 50-1 of IFAC PapersOnLine, 2017, p. 1772?1783 (2017).
[14] M. H. T. Hack, Decision problems for Petri nets and Vector Addition Systems, Tech. Rep. Computation Structures Group Memo 95. Project MAC, Laboratory for Computer Science, M.I.T., Cambridge, MA (1974).
[15] J. L. Peterson, Petri nets, ACM Computing Surveys 9 (3) (1977) 223–252 (1977).
[16] A. W. Holt, F. Commoner, Events and Conditions, Record Project MAC Conference on Concurrent Systems Parallel Computation (1970) 3–52 (1970).
[17] T. Agerwala, M. Flynn, Comments on Capabilities, Limitations and "Correctness" of Petri Nets, ACM SIGARCH Comp. Architecture News 2 (4) (1973) 81–86 (1973).
[18] K. Jensen, G. Rozenberg (Eds.), High-level Petri Nets, Springer, 1991 (1991).
[19] H. Alla, P. Ladet, J. Martínez, M. Silva, Modelling and Validation of Complex Systems modeled by Colored Petri Nets, in: G. Rozenberg (Ed.), Advances in Petri Nets 1984, Vol. 609 of LNCS, Springer-Verlag, 1985, pp. 15–31 (1985).
[20] A. Valmari, Stubborn Sets for Reduced State Space Generation, in: G. Rozenberg (Ed.), Advances in Petri Nets 1990, Vol. 483 of LNCS, Springer, 1991, pp. 491–515 (1991).
[21] P. H. Starke, Reachability analysis of Petri nets using symmetries, Systems Analysis Modelling Simulation 8 (4-5) (1991) 293–303 (1991).
[22] Z. Ma, Y. Tong, Z. Li, A. Giua, Basis Marking Representation of Petri Net Reachability Spaces and its Application to the Reachability Problem, IEEE Trans. on Automatic Control (2016).
[23] A. Giua, X. Xie, Control of safe ordinary Petri nets using unfolding, Discrete Event Dynamic Systems 15 (2005) 349–373 (2005). doi:10.1007/s10626-017-0238-9.
[24] J. Esparza, K. Heljanko, Unfoldings: A Partial-Order Approach to Model Checking, 1st Edition, EATCS Monographs in Theoretical Computer Science, Springer-Verlag, 2008 (2008).
[25] G. Chiola, C. Dutheillet, G. Franceschinis, S. Haddad, A symbolic reachability graph for coloured petri nets, Theoretical Computer Science 176 (1-2) (1997) 39–65 (1997).
[26] H. Baker, Equivalence problems in Petri nets, Tech. Rep. Computation Structures Group. Project MAC, Lab. for Computer Science, M.I.T., Cambridge, MA (1973).
[27] M. Jantzen, Language Theory of Petri Nets, in: W. Brauer, W. Reisig, G. Rozenberg (Eds.), Advances in Petri Nets 1986, Vol. 254 of LNCS, Springer, 1991, pp. 397–412 (1991).
[28] G. Vidal-Naquet, Deterministic Petri net languages, in: C. Girault, W. Reisig (Eds.), Application and Theory of Petri Nets, Springer, 1982, pp. 108–202 (1982).
[29] G. Berthelot, Checking Properties of Nets Using Transformations, in: G. Rozenberg (Ed.), Advances in Petri Nets 1985, Vol. 222 of LNCS, Springer, 1986, pp. 19–40 (1986).
[30] K. Lautenbach, H. A. Schmid, Use of Petri nets for proving correctness of concurrent process systems, in: IFIP Congress, 1974, pp. 187–191 (1974).
[31] T. Murata, Circuit Theoretic Analysis and Synthesis of Marked Graphs, IEEE Trans. on Circuits and Systems 24 (7) (1977) 400–405 (1977).
[32] J. Sifakis, Structural Properties of Petri Nets, in: J. Winkowski (Ed.), Mathematical Foundations of Computer Science 1978, Springer, 1978, pp. 474–483 (1978).
[33] G. Memmi, G. Roucairol, Linear Algebra in Net Theory, in: Brauer (9), pp. 213–223 (1980).
[34] M. Silva, J. M. Colom, On the Computation of Structural Synchronic Invariants in P/T Nets, in: G. Rozenberg (Ed.), Advances in Petri Nets 1988, Vol. 340 of LNCS, Springer, 1988, pp. 387–417 (1988).
[35] J. Colom, M. Silva, Improving the linearly based characterization of P/T nets, in: G. Rozenberg (Ed.), Advances in Petri Nets'90, Vol. 483 of LNCS, Springer-Verlag, 1991, pp. 113–145 (1991).
[36] M. Silva, J. Campos, Performance evaluation of deds with conflicts and synchronizations: Net-driven decomposition techniques, in: 4th IEE Int. Workshop on DEDS, WODES'98, 1998, pp. 398–413 (1998).
[37] G. Chiola, C. Dutheillet, G. Franceschinis, S. Haddad, Stochastic well-formed coloured nets for symmetric modelling applications, IEEE Trans. on Computers 42 (11) (1993) 1343–1360 (1993).
[38] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, G. Franceschinis, Modelling with Generalized Stochastic Petri Nets, 1st Edition, John Wiley & Sons, Inc., New York, NY, USA, 1994 (1994).
[39] G. Balbo, M. Silva (Eds.), 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) (1998).
[40] M. Malhotra, K. Trivedi, Dependability modeling using Petri nets, IEEE Trans. on Reliability 44 (3) (1995) 428–440 (1995).
[41] B. P. Zeigler, Theory of Modeling and Simulation, John Wiley, 1976 (1976).
[42] B. P. Zeigler, A. Muzy, From discrete event simulation to discrete event specified systems (devs), IFAC PapersOnLine 50 (1) (2017) 3039–2044 (2017).
[43] R. David, H. Alla, Du Grafcet aux réseaux de Petri, Hermes, 1989 (1989).
[44] Petri Nets World, *Petri Nets Tool Database* (2017).
URL `https://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/db.html`
[45] J. L. Peterson, Petri Net Theory and the Modeling of Systems, Prentice-Hall, Upper Saddle River, NJ, 1981 (1981).
[46] P. Starke, Petri-Netze, VeB Deutscher Verlag der Wissenschaften, Berlin, 1981 (1981).
[47] W. Reisig, Petrinetze. Eine Einführung, Springer-Verlag, 1982 (1982).
[48] W. Reisig, Petri Nets. An introduction, Springer-Verlag, 1985 (1985).
[49] G. W. Brams, Réseaux de Petri: Théorie et Pratique, Masson, 1983 (1983).
[50] M. Silva, Las Redes de Petri: en la Automatica y la Informática, Madrid, Ed. AC, 1985 (1985).
[51] B. Krogh, Controlled Petri nets and maximally permissive feedback logic, in: Proceedings of 25th Annual Allerton Conference, University of Illinois, Urbana, 1987 (1987).
[52] S. Balemi, P. Kozák, R. Smedinga (Eds.), Discrete Event Systems: Modeling and Control, Vol. 13 of Progress in Systems and Control Theory, Birkhaüsen, 1993 (1993).
[53] P. Ramadge, W. Wonham, The control of discrete event systems, Procs. of the IEEE 77 (1) (1989) 81–98 (Jan 1989).
[54] W. M. Wonham, K. Cai, K. Rudie, Supervisory Control of Discrete Event Systems: A Brief History, Annual Reviews in Control 45 (2018) ¿–? (2018).
[55] L. Holloway, B. Krogh, A. Giua, A survey of Petri net methods for controlled discrete event systems, Discrete Event Dynamic Systems 7 (2) (1997) 151–190 (1997).
[56] A. Giua, F. DiCesare, Blocking and controllability of Petri nets in supervisory control, IEEE Trans. on Automatic Control 39 (4) (1994) 818–823 (Apr 1994).
[57] A. Giua, F. DiCesare, Decidability and closure properties of weak Petri net languages in supervisory control, IEEE Trans. on Automatic Control 40 (5) (1995) 906–910 (May 1995).
[58] R. Kumar, L. Holloway, Supervisory control of deterministic Petri nets with regular specification languages, IEEE Trans. on Automatic Control 41 (2) (1996) 245–249 (Feb 1996).
[59] R. Sreenivas, B. Krogh, On Petri net models of infinite state supervisors, IEEE Trans. on Automatic Control 37 (2) (1992) 272–277 (Feb 1992).
[60] A. Giua, F. DiCesare, Supervisory design using Petri nets, in: 30th IEEE Conference on Decision and Control, 1991, pp. 92–97 (1991).
[61] A. Ghaffari, N. Rezg, X. Xie, Design of a live and maximally permissive Petri net controller using the theory of regions, IEEE Trans. on Robotics and Automation 19 (1) (2003) 137–141 (2003).
[62] R. 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) (1997) 928–945 (Jul 1997).
[63] K. He, M. Lemmon, Liveness-enforcing supervision of bounded ordinary Petri nets using partial order methods, IEEE Trans. on Automatic Control 47 (7) (2002) 1042–1055 (Jul 2002).
[64] X. Xie, A. Giua, Counterexamples to ”Liveness-enforcing supervision of bounded ordinary Petri nets using partial-order methods”, IEEE Trans. on Automatic Control 49 (7) (2004) 1217–1219 (Jul 2004).
[65] L. Holloway, B. Krogh, Synthesis of feedback control logic for a class of controlled Petri nets, IEEE Trans. on Automatic Control 35 (5) (1990) 514–523 (May 1990).
[66] L. Holloway, B. Krogh, Synthesis of feedback control logic for discrete manufacturing systems, Automatica 27 (4) (1991) 641–651 (Apr 1991).
[67] Y. Li, W. Wonham, Control of vector discrete-event systems ii - controller synthesis, IEEE Trans. on Automatic Control 39 (3) (1994) 512–531 (Mar 1994).
[68] A. Giua, F. DiCesare, M. Silva, Generalized Mutual Exclusion Constraints on Nets with Uncontrollable Transitions, in: IEEE International Conference on Systems, Man and Cybernetics, 1992, pp. 974–979 (1992).
[69] K. Yamalidou, J. Moody, M. Lemmon, P. Antsaklis, Feedback control of Petri nets based on place invariants, Automatica 32 (1) (1996) 15–28, a preliminary version of this paper was presented at CDC 1994 (Jan 1996).
[70] J. Moody, P. Antsaklis, Petri net supervisors for des with uncontrollable and unobservable transitions, IEEE Trans. on Automatic Control 45 (3) (2000) 462–476 (Mar 2000).
[71] M. Iordache, P. Antsaklis, Synthesis of supervisors enforcing general linear constraints in Petri nets, IEEE Trans. on Automatic Control 48 (11) (2003) 2036–2039 (Nov 2003).

[72] F. Basile, P. Chiacchio, A. Giua, An optimization approach to Petri net monitor design, IEEE Trans. on Automatic Control 52 (2) (2007) 306–311 (Feb 2007).

[73] A. Dideban, H. Alla, Reduction of constraints for controller synthesis based on safe Petri nets, Automatica 44 (7) (2008) 1697–1706 (2008).

[74] J. Luo, K. Nonami, Approach for transforming linear constraints on Petri nets, IEEE Trans. on Automatic Control 56 (12) (2011) 2751–2765 (Dec 2011).

[75] F. Basile, R. Cordone, L. Piroddi, Integrated design of optimal supervisors for the enforcement of static and behavioral specifications in Petri net models, Automatica 49 (1) (2013) 3432–3439 (2013).

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

[77] F. Commoner, Deadlocks in Petri nets, Tech. Rep. Report CA-7206-2311, Laboratory for Computer Science, M.I.T., Wakefield, MA (1972).

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

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

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

[81] J. Ezpeleta, F. Tricas, F. García-Vallés, J. Colom, A banker's solution for deadlock avoidance in FMS with flexible routing and multiresource states, IEEE Trans. on Robotics and Automation 18 (2002) 621–625 (August 2002).

[82] M. Iordache, J. Moody, P. Antsaklis, Synthesis of deadlock prevention supervisors using petri nets, IEEE Trans. on Robotics and Automation 18 (1) (2002) 59–68 (2002).

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

[84] Z. Li, M. 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) (2004) 38–51 (Jan 2004).

[85] 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) (2002) 192–208 (2002).

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

[87] M. Silva, E. Teruel, Petri nets for the design and operation of manufacturing systems, European Journal of Control 3 (3) (1997) 182–199 (1997).

[88] J. Baer, A survey of some theoretical aspects of multiprocessing, ACM Computing Surveys 5 (1) (1973) 31–80 (1973).

[89] J. Carlier, P. Chretienne, C. Girault, Modelling scheduling problems with timed Petri nets, in: G. Rozenberg (Ed.), Advances in Petri Nets, Vol. 188 of LNCS, Springer-Verlag, 1985, pp. 62–82 (1985).

[90] J. Carlier, P. Chretienne, Timed Petri net schedules, in: Advances in Petri Nets, Vol. 340 of LNCS, Springer-Verlag, 1988, pp. 62–84 (1988).

[91] J. Magott, New NP-complete problems in performance evaluation of concurrent systems using petri nets, IEEE Trans. on Software Engineering 13 (5) (1987) 578–581 (1987).

[92] J. Martinez, P. Muro, M. Silva, Modeling, validation and software implementation of production systems using high level Petri nets, in: IEEE Int. Conf. on Robotics and Automation, IEEE, 1987 (1987).

[93] D. Lee, F. DiCesare, Scheduling flexible manufacturing systems using Petri nets and heuristic search, IEEE Trans. on Robotics and Automation 10 (2) (1994) 123–132 (1994).

[94] G. Tuncel, G. Bayhan, Applications of Petri nets in production scheduling: a review, Int. Jour. of Advanced Manufacturing Technology 34 (7-8) (2007) 762–773 (2007).

[95] M. Zhou, Petri Nets in Flexible and Agile Automation, The Springer International Series in Engineering and Computer Science, Springer US, 1995 (1995).
URL `https://books.google.it/books?id=C-tSAAAAMAAJ`

[96] M. Zhou, M. Jeng, Modeling, analysis, simulation, scheduling, and control of semiconductor manufacturing systems: A Petri net approach, IEEE Trans. on Semiconductor Manufacturing 11 (3) (1998) 333–357 (1998).

[97] R. Srinivasan, Modeling and performance analysis of cluster tools using Petri nets, IEEE Trans. on Semiconductor Manufacturing 11 (3) (1998) 394–403 (1998).

[98] H.-Y. Lee, T.-E. Lee, Scheduling single-armed cluster tools with reentrant wafer flows, IEEE Trans. on Semiconductor Manufacturing 19 (2) (2006) 226–240 (2006).

[99] N. Wu, C. Chu, F. Chu, M. Zhou, A Petri net method for schedulability and scheduling problems in single-arm cluster tools with wafer residency time constraints, IEEE Trans. on Semiconductor Manufacturing 21 (2) (2008) 224–237 (2008).

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

[101] Y. Ru, C. Hadjicostis, Fault diagnosis in discrete event systems modeled by partially observed Petri nets, Discrete Event Dynamic Systems 19 (4) (2009) 551–575 (2009).

[102] Y. Tong, Z. Li, A. Giua, On the equivalence of observation structures for Petri net generators, IEEE Trans. on Automatic Control 61 (9) (2013) 2448–2462 (2013).

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

[104] A. Giua, C. Seatzu, F. Basile, Observer-based state feedback control of timed Petri nets with deadlock recovery, IEEE Trans. on Automatic Control 49 (1) (2004) 17–29 (2004).

[105] A. Giua, D. Corona, C. Seatzu, State estimation of lambda-free labeled Petri nets with contact-free nondeterministic transitions, Discrete Event Dynamic Systems 15 (1) (2005) 85–108 (2005).

[106] D. Corona, A. Giua, C. Seatzu, Marking estimation of Petri nets with silent transitions, IEEE Trans. on Automatic Control 52 (9) (2007) 1695–1699 (2007).

[107] Y. Ru, C. Hadjicostis, Sensor selection for structural observability in discrete event systems modeled by Petri nets, IEEE Trans. on Automatic Control 55 (8) (2010) 1751–1764 (2010).

[108] L. Li, C. Hadjicostis, Least-cost transition firing sequence estimation in labeled Petri nets with unobservable transitions, IEEE Trans. on Automation Science and Engineering 8 (2) (2011) 394–403 (2011).

[109] L. Li, C. Hadjicostis, Minimum initial marking estimation in labeled Petri nets, IEEE Trans. on Automatic Control 58 (1) (2013) 198–203 (2013).

[110] M. Cabasino, C. Hadjicostis, C. Seatzu, Probabilistic marking estimation in labeled Petri nets, IEEE Trans. on Automatic Control 60 (2) (2015) 528–533 (2015).

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

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

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

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

[115] V. Sreenivas, M. Jafari, Fault detection and monitoring using time Petri nets, IEEE Trans. Systems, Man and Cybernetics 23 (4) (1993) 1155–1162 (1993).

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

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

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

[119] D. Lefebvre, C. Delherm, Diagnosis of DES with Petri net models, IEEE Trans. on Automation Science and Engineering 4 (1) (2007) 114–118 (2007).

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

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

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

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

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

[125] F. Basile, M. Cabasino, C. Seatzu, Diagnosability analysis of labeled time Petri net systems, IEEE Trans. on Automatic Control 62 (3) (2017) 1384–1396 (April 2017).

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

[127] E. Badouel, L. Bernardinello, P. Darondeau, Petri Net Synthesis, Springer, 2015 (2015).

[128] B. Van Dongen, A. De Medeiros, L. Wen, Process mining: Overview and outlook of Petri net discovery algorithms, in: Trans. on Petri Nets and Other Models of Concurrency II, Springer, 2009, pp. 225–242 (2009).

[129] E. Badouel, P. Darondeau, Theory of regions, in: Lectures on Petri Nets I: Basic Models, Springer, 1998, pp. 529–586 (1998).

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

[131] M. Dotoli, M. Fanti, A. Mangini, Real time identification of discrete event systems using Petri nets, Automatica 44 (5) (2008) 1209–1219 (2008).

[132] Y. Wu, C. Hadjicostis, Algebraic approaches for fault identification in discrete-event systems, IEEE Trans. on Automatic Control 50 (12) (2005) 2048–2055 (Dec 2005).

[133] M. Cabasino, A. Giua, C. Hadjicostis, C. Seatzu, Fault model identification and synthesis in petri nets, Discrete Event Dynamic Systems 25 (3) (2015) 419–440 (Sep 2015). doi:10.1007/s10626-014-0190-x.
URL `https://doi.org/10.1007/s10626-014-0190-x`

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

[135] R. David, H. Alla, Continuous Petri Nets, in: Proc. of the 8th European Workshop on Application and Theory of Petri Nets, Zaragoza, 1987, pp. 275–294 (1987).

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

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

[138] L. Recalde, E. Teruel, M. Silva, Autonomous Continuous P/T systems, in: S. Donatelli, J. Kleijn (Eds.), Application and Theory of Petri Nets 1999, Vol. 1639 of LNCS, Springer, 1999, pp. 107–126 (1999).

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

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

[141] A.-L. Meyer, Discontinuity induced bifurcations in timed continuous Petri nets, in: Int. Workshop on Discrete Event Systems (WODES'12), 2012, pp. 28–33 (2012).

[142] M. Navarro-Gutiérrez, A. Ramírez-Trevino, M. Silva, Homothecy, bifurcations, continuity and monotonicity in timed continuous Petri nets under infinite server semantics, Nonlinear Analysis: Hybrid Systems 26 (2017) 48–67 (2017).

[143] E. Fraca, J. Júlvez, 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.

[144] C. Vázquez, 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) (2012) 641–653 (2012).

[145] M. Beccuti, E. Bibbona, A. Horváth, R. Sirovich, A. Angius, G. Balbo, Analysis of Petri Net Models through Stochastic Differential Equations, in: G. Ciardo, E. Kindler (Eds.), Application and Theory of Petri Nets and Concurrency, Vol. 8489 of LNCS, Springer, 2014, pp. 273–293 (2014).

[146] D. Lefebvre, E. Leclerq, N. E. Akchioui, L. Khalij, E. D. Cursi, A geometric approach for the homothetic approximation of stochastic Petri nets, in: Int Workshop on Discrete Event Systems (WODES'10), Berlin, 2010, pp. 245–250 (2010).

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

[148] C. Mahulea, C. Seatzu, M. Cabasino, 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) (2012) 970–984 (2012).

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

[150] C. Seatzu, M. Silva, J. Schuppen (Eds.), Control of Discrete-Event Systems. Automata and Petri Net Perspectives, Vol. 433 of LNCIS, Springer, 2013 (2013).

[151] Y. Wardi, A. Giua, C. Seatzu, IPA for continuous stochastic marked graphs, Automatica 49 (5) (2013) 1204–1215 (2013).

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

[153] J. L. Bail, H. Alla, R. David, Hybrid Petri nets, in: Proc. 1st Int. European Control Conference (ECC91), Grenoble, France, 1991, pp. 1472–1477 (1991).

[154] K. Trivedi, V. G. Kulkarni, FSPNs: Fluid Stochastic Petri Nets, in: M. Ajmone Marsan (Ed.), Application and Theory of Petri Nets 1993, Vol. 691 of LNCS, Springer, Berlin, 1993, pp. 24–31 (1993).

[155] R. Alur, D. Dill, The theory of timed automata., Theoretical Computer Science 126 (1994) 183–235 (1994).

[156] C. Valentin-Roubinet, Modeling of hybrid systems: DAE supervised by Petri nets, in: 3rd IFAC Int. Conference

on Automation of Mixed Processes (ADPM98), Int. Federation of Automatic Control, Papers On Line, 1998, pp. 142–149 (1998).
[157] R. Champagnat, P. Esteban, H. Pingaud, R. Valette, Petri net based modeling of hybrid systems., Computers in Industry 36 (1-2) (1998) 139–146 (1998).
[158] I. Demongodin, N. T. Koussoulas, Differential Petri nets: Representing continuous systems in a discrete-event world, IEEE Trans. on Automatic Control 43 (4) (1998) 573–579 (1998).
[159] A. Giua, E. Usai, Modelling hybrid systems by high-level Petri nets, European Journal of Automation APII-JESA 32 (9–10) (1998) 1209–1231 (1998).
[160] P. J. Antsaklis, X. Koutsouko, J. Zaytoon, On Hybrid Control of Complex Systems: A Survey, European Journal of Automation (APII-JESA) 32 (9-10) (1998) 1023–1045 (1998).
[161] A. Di Febbraro, A. Giua, G. Menga, editors, Special Issue on Hybrid Petri Nets, Discrete Event Dynamic Systems 11 (1-2) (2001).
[162] F. Balduzzi, A. Giua, G. Menga, First-Order Hybrid Petri nets: A model for optimization and control, IEEE Trans. on Robotics and Automation 16 (4) (2000) 382–399 (2000).
[163] I. Demongodin, Generalised Batches Petri Net: Hybrid model for high speed systems with variable delays, Discrete Event Dynamic Systems 11 (1-2) (2001) 137–162 (2001).
[164] I. Demongodin, 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, 2010, pp. 4–9 (2010).
[165] C. Vázquez, M. Silva, Stochastic hybrid approximations of markovian Petri nets, IEEE Trans. on Systems, Man, and Cybernetics: Systems 45 (9) (2015) 1231–1244 (2015).
[166] F. Balduzzi, A. Giua, C. Seatzu, Modelling and simulation of manufacturing systems with first-order hybrid Petri nets, Int. Jour. of Production Research 39 (2) (2001) 255–282 (2001).
[167] M. Allan, H. Alla, Modeling and simulation of an electronic component manufacturing system using hybrid Petri nets, IEEE Trans. on Semiconductor Manufacturing 11 (3) (1998) 374–383 (1998).
[168] X. Zha, S. Lim, Assembly/disassembly task planning and simulation using expert Petri nets, Int. Jour. of Production Research 38 (15) (2000) 3639–3676 (2000).
[169] T. Gu, P. Bahri, A survey of petri net applications in batch processes, Computers in Industry 47 (2002) 99–111 (2002).
[170] A. Di Febbraro, D. Giglio, N. Sacco, Urban traffic control structure based on hybrid Petri nets, IEEE Trans. on Intelligent Transportation Systems 5 (4) (2004) 224–237 (2004).
[171] M. Dotoli, M. Fanti, G. Iacobellis, A. Mangini, A first-order hybrid Petri net model for supply chain management, IEEE Trans. on Semiconductor Manufacturing 6 (4) (2009) 744–758 (2009).
[172] J. Komenda, S. Lahaye, J. Boimond, T. van den Boom, Max-Plus Algebra in the History of Discrete Event Systems, Annual Reviews in Control 45 (2018) ¿–? (2018).
[173] L. Grüne, S. Hirche, O. Junge, P. Koltai, D. Lehmann, J. Lunze, A. Molin, R. Sailer, M. Sigurani, C. Stöcker, F. Wirth, Event-based control, in: J. Lunze (Ed.), Control Theory of Digitally Networked Dynamic Systems, Springer, 2014, pp. 169–261 (2014).
[174] T. Murata, Petri Nets: Properties, Analysis and Applications, Procs. of the IEEE 77 (4) (1989) 541–580 (1989).
[175] W. Van Der Aalst, The application of Petri nets to workflow management, Vol. 8, 1998, pp. 21–66 (1998).
[176] A. Bemporad, M. Morari, Control of systems integrating logic, dynamics, and constraints, Vol. 35, 1999, pp. 407–427 (1999).
[177] S. Lafortune, F. Lin, C. Hadjicostis, On the History of Diagnosability and Opacity in Discrete Event Systems, Annual Reviews in Control 45 (2018) ¿–? (2018).

Appendix: Petri nets in the Automatic Control literature: A biased bibliometric search

In this appendix we collect some factual data about the development of Petri Nets literature within the area of Systems Theory and Automatic Control.

The first conceptually "biased" decision has been to limit our analysis to a subset of well established journals. In fact, for practical reasons we were unable to consider conference publications or technical reports, even if this material was very relevant especially during the first three decades of the discipline, and remains very important today. As an example, here we do not consider the *Advances in Petri Nets*, a subseries of *Lecture Notes in Computer Science* edited by Grezgorz Rozenberg which was extremely important from 1984 and for almost two decades. Analogously, it is not easy to collect data on books in the field, even if one can built a list of one hundred volumes. Let us also remark that this research area started with yet another type of publication, i.e., a Ph.D. thesis for which in 2008 C. A. Petri received the IEEE Computer Pioneer award "for establishing Petri net theory in 1962, which not only was cited by hundreds of thousands of scientific publications but also significantly advanced the fields of parallel

		PN			*DES*			*Both*	*NP_PN /*
		NP	NC	NC/NP	NP	NC	NC/NP	NP	*NP_DES*
ARC	*Annual Reviews in Control*	4	190	47.5	12	394	32.8	3	0.33
AJC	*Asian Journal of Control*	36	320	8.9	25	118	4.7	10	1.44
AUT	*Automatica*	55	2,247	40.9	138	5,041	36.5	25	0.40
CEP	*Control Engineering Practice*	46	773	16.8	62	998	16.1	19	0.74
DEDS	*Discrete Event Dynamic Systems*	86	1,855	21.6	242	4,735	19.6	37	0.36
EJC	*European Journal of Control*	2	62	31.0	22	168	7.6	0	0.09
TAC	*IEEE Trans. Automatic Control*	121	4,116	34.0	390	12,826	32.9	72	0.31
TASE	*IEEE Trans. Automation Science & Engin.*	100	3,211	32.1	123	2,328	18.9	50	0.81
TCSi	*IEEE Trans. Circuits and Systems I*	4	159	39.8	2	491	245.5	0	2.00
TCST	*IEEE Trans. Control Systems Technology*	29	476	16.4	49	1,555	31.7	13	0.59
TRA	*IEEE Trans. Robotics & Automation*	59	5,472	92.7	30	1,334	44.5	10	1.97
TSMC	*IEEE Trans. Systems, Man & Cyb.*	30	1,349	45.0	19	473	24.9	2	1.58
TSMCa	*IEEE Trans. Systems, Man & Cyb. Part A*	130	5,739	44.1	71	2,209	31.1	38	1.83
TSMCb	*IEEE Trans. Systems, Man & Cyb. Part B*	46	1,308	28.4	30	1,022	34.1	9	1.53
TSMCc	*IEEE Trans. Systems, Man & Cyb. Part C*	35	1,722	49.2	18	767	42.6	10	1.94
TSMCsys	*IEEE Trans. Systems, Man & Cyb. Systems*	55	642	11.7	31	436	14.1	21	1.77
TIE	*IEEE Trans.Industrial Electronics*	34	1,559	45.9	17	867	51.0	5	2.00
IJC	*International Journal of Control*	20	110	5.5	77	744	9.7	8	0.26
JCSC	*Journal of Circuits, Systems and Computers*	12	2,038	169.8	5	231	46.2	1	2.40
NAHS	*Nonlinear Analysis: Hybrid Systems*	28	291	10.4	20	199	10.0	14	1.40
PE	*Performance Evaluation*	82	2,069	25.2	25	440	17.6	8	3.28
PIEEE	*Proceedings of the IEEE*	9	6,575	730.6	24	3,682	153.4	2	0.38
		1,023	42,283	41.3	1,432	41,058	28.7	357	0.71

Table .1
Total number of published papers in selected journals (NP), total number of citations (NC) and average number of citations per item (NC/NP).

and distributed computing.” Finally, a rather complementary set of references, with a Computer Science bias, can be accessed in the so called *The Petri Nets Bibliography*: http://www.informatik.uni-hamburg.de/TGI/pnbib/.

The analysis sketched in this appendix is based on data collected in March 2018 from Scopus. We considered publications appeared in the journals that are listed in Table 7.1 with their acronym. Most of them can be labeled as primarily focusing on Systems Theory or Automatic Control, but it should be recognized that many other journals that are not mentioned here, such as the *IEEE Trans. on Software Engineering* or *Theoretical Computer Science* also publish since the 80s a significant number of papers on Systems Theory.

We have searched for papers that contain “Petri” or “discrete event” in the fields *Title, Abstract and Keywords.* In the following we call these papers *PN papers* and *DES papers*, respectively. We are aware that a classification based on such a search is necessarily imprecise. As an example, if we limit our analysis to the journal *Discrete Event Dynamic Systems* — that arguably contains only DES papers — out of 510 papers published since 1991 and up to the first issue of 2018, only 242 are retrieved by the query DES. Similarly, of out 86 papers retrieved in the same journal with the query PN, only 37 also belong to the query DES.

As an additional example, let us consider the three most cited papers in each query. Unsurprisingly, among the the most cited PN papers some are not so centered on AC. Such is the case for the paper by (174) [6,226 citations], which is a survey/tutorial from a broad engineering point of view, and for the paper by (175) [1,744 citations], which is a survey on workflow processes appealing to a much larger audience. Only the third one, by (80) [843 citations], can be considered a full-fledged AC paper and it is considered a seminal work for deadlock control. Similarly, among the most cited DES papers the first one, co-authored by (176) [1,794 citations], belongs to a different and larger AC subdomain, namely hybrid systems. The other two by (53) [1,674 citations] and by (117) [932 citations] are central to the DES domain, as discussed in the two historical perspectives by (54) and (177).

The paper by (53) was published in the January issue of the *Proceedings of the IEEE*, as part of a special issue devoted to DES. The issue contained several seminal papers which contributed to give the area of DES a full recognition as a subdomain of Control Systems. The special issue, however, paid little attention to Petri nets, at a time in which they were still largely unknown within the AC community. By a fortuitous coincidence, however, the paper by (174) appeared just three months later on the same journal.

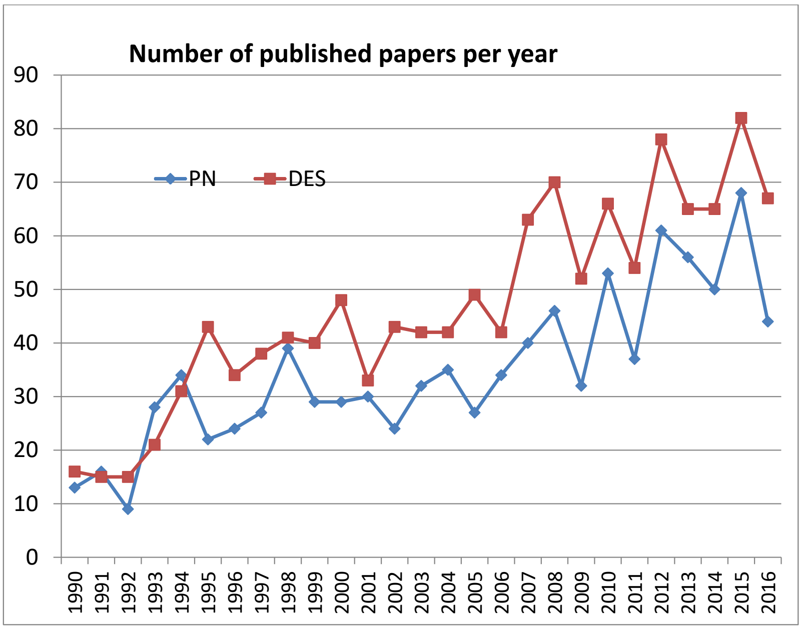

Fig. .1. Trend of the number of published papers per year in selected journals.

In Table 7.1 we analyze papers published in the selected journals. The table shows for each journal and each topic the number of papers (NP), the total number of citations received (NC) and the average number of citations per item (NC/NP). The last but one column shows how many papers belong to both groups, PN papers and DES papers. The last column shows the ratio between the number of PN papers and the number of DES papers for each journal.

We can notice that the ratio [number of PN papers] / [number of DES papers] is 1,023/1,432, i.e, roughly 71%. If, however, we assume that all PN papers are also DES paper and compute the ratio [number of PN papers] / [number of PN or DES papers] we obtain a value of 49%.

Among the selected journal those that can be labeled as Automatic Control journals are: DEDS; the publications of the IEEE Control Systems Society TAC and TCST; the IFAC publications AUT, CEP, NAHS and ARC; IJC, EJC and AJC. All other journals belong to the broader area of Systems Theory. From Table 7.1 one can observe that all Automatic Control journals have a ratio PN papers versus DES papers less than 1.5 while all other journals (except the PIEEE) have a higher ratio.

Figure .1 shows the number of PN papers and DES papers published each year from 1990 to 2016. For both topics the trend is very similar. We can see that during this period the number of papers has increased on the average. In 2015 (when a local maximum was reached) 82 DES papers and 68 PN papers have been published in the selected journals.

Figure .2 shows the trend of the 5 Year Impact Factor (IF) of papers published in the selected journals from 1993 to 2015 (the data was collected from Web of Sciences, because the IF is based on citations measured on this database). One can observe that starting from 2009 PN papers are having a higher impact than DES papers. This is consistent with the data in Table 7.1, where we can see that the average number of citations per item (NC/NP) is higher for PN papers with a ration [average number of citations PN papers] / [average number of citations DES papers] equal to 41.3/28.7, i.e., roughly 144%. This could be explained by the larger audience for PN papers, that may also include researchers that do not belong to the Automatic Control community.

A final remark concerns the significance of the presented data. One should keep in mind that using different bibliographic databases the numerical values may significantly change, although the relative ordering should not change. As an example, a previous analysis made in 2015 with data collected from Web of Sciences relative to the same journals, showed a ratio [number of PN papers] / [number of DES papers] equal to 996/1,226 (i.e., 81% as opposed to 71%) and a ratio [average number of citations PN papers] / [average number of citations DES papers] equal to 21.2/17 (i.e., 125% as opposed to 144%).

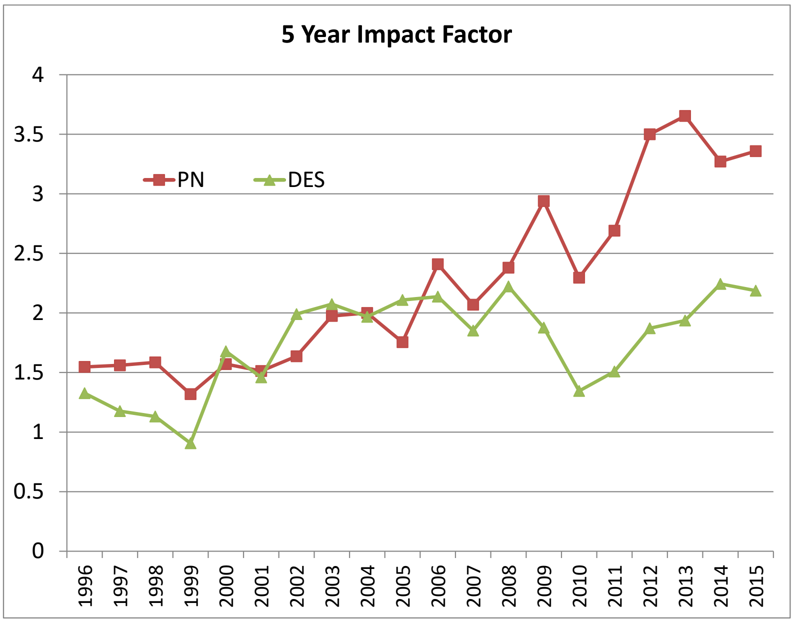

Fig. .2. Trend of the 5 Year Impact Factor of papers published in selected journals.