Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 1772–1783

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

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

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

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

---

1. PRELIMINARY OVERVIEW

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

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

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

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

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

---

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

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

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

10.1016/j.ifacol.2017.08.156

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

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

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

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

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

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

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

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

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

---

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

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