50 years after the PhD thesis of Carl Adam Petri: A perspective

Manuel Silva*

* Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain (e-mail: silva@unizar.es)

Abstract: Half a century of a work defining a landmark in Discrete Event Dynamic Systems theory is worth underlining. This invited contribution, of festive character, albeit I fear inadequate, is a modest tribute to the memory and work of Carl Adam Petri. It was considered convenient to remember some of the personal contacts we had with him. The price is a certainly biased view.

Keywords: Carl Adam Petri, Condition/Event net, Petri Net (PN), PN paradigm, Discrete Event Dynamic System (DEDS), Fluidization

1. INTRODUCTION

Science and Technology are social constructions. Nevertheless, in their development some people contribute in outstanding ways, as recognized by Isaac Newton in a letter to Robert Hooke (1676): “If I have seen further it is by standing on ye sholders of Giants” 1. The recognition of their achievements is sometimes partially done by giving his/her name to some measurement unit, universal constant, algorithm, etc. For example, if dealing with measurement units, in the International System of Units (SI) we have: the newton (named after Isaac Newton, 1642, 1727), the coulomb (after Charles-Augustin de Coulomb, 1736-1806), the volt (after Alessandro Volta, 1745-1827), the ampere (after André Marie Ampère, 1775-1836), or the farad (after Michael Faraday, 1791-1867). Looking at temperatures there are the well-known scales of Celsius, Fahrenheit, Reaumur, Rankine and Kelvin (this last defines a universal constant: the 0 K). In other opportunities, researcher’s names are given to some special facts, like the force of Coriolis, the constant or number of Avogadro, the algorithm of Dijkstra (shortest paths), the Wiener or the Kalman filters, or the Forrester diagrams. In some cases, the equation, algorithm, etc, receives two or more names. For example, among basic models of predator-prey problems is the so called Lotka-Volterra equations; in the computation domain, the Floyd-Warshall algorithm is also known as Roy-Warshall or Roy-Floyd algorithm. In this well-intentioned dynamics “excesses” are sometimes done. For example, often we hear about the “Watt governor”, while making reference to the classical centrifugal or “flyball” governor. By no means can we doubt about the outstanding contributions of James Watt to the steam machine and its clear consequences on the Industrial revolution, but this governor, used by Watt, was not invented by him; it was previously patented (Mayr, 1970). On the contrary, even if the paternity of a discovery is clear, the name of the discoverer is not usually on it. For example, George Dantzig is properly acclaimed as the “father of linear programming”, but his algorithm is just called the “simplex method” (1948).

In relatively very few cases, the name of a researcher is given to a theory for an entire subfield. For example, we speak of Markov Chains (MC) after Andrei Markov (1856-1922) 2. Of course, the theory of (semi-)Markov Chains is a collective work, not a personal one, but the Russian mathematician did play the role of pioneer. Analogous is the case with the so called Petri Nets (PNs), a system theory initially inspired by Carl Adam Petri (1926-2010). The first stone of this construction, in which the so called Petri nets are not defined, is his PhD dissertation (Petri, 1962) 3. Initially, PNs were considered as part of Computer Science (CS), but very quickly they began to be employed also in Automatic Control (AC) for automation; last but not least, PNs were incorporated to the background of Operations Research (OR). Therefore, PNs are perceived as part of the Discrete Event Dynamic Systems (DEDS) theory, at the intersection of CS, AC and OR. Moreover, fluid (or continuous) and different kinds of hybrid PNs are being extensively studied today.

If the mathematics for continuous dynamic “views” of systems, particularly for control, go back more than three centuries (Sussmann and Willems, 1997), the formalization of discrete event “views” of dynamic systems is much more recent. Even if several precedents exist (Erlang, Shannon, Huffman, Moore, Mealy, etc.), roughly speaking it can be said that such “views” were really developed during the second half of the past century. Moreover, in the framework of computer-based simulation, a subfield in computer engineering, there were important initiatives in the so called “discrete event simulation” (Fishman, 1973; Zeigler, 1976). In particular, in Zeigler (1976) the Discrete Event System Specification (DEVS) was introduced “to provide a

---

1 In modern English: “If I have seen further it is by standing on the shoulders of giants”.

2 As it is very well-known, it is a sequential stochastic system that enjoys the memoryless property. In fact, “memorylessness” is the so called “Markov property.”

3 Translation into English: (Petri, 1966).
formal basis for specifying the models expressible within discrete event simulation languages (SIMSCRIPT, SIMULA, GPSS, etc.)" (Zeigler, 1984).

The expression Discrete Event Dynamic System (DEDS) started to be used in 1980, by Y. C. Ho and his team at Harvard, becoming popular by the end of the same decade (see, for example, (Ho, ed., 1989)). Like Le Bourgeois Gentilhomme of Molière, more or less suddenly, many of us, users or developers on topics related to automata, Markov chains, queueing networks, Petri nets, process algebras, max-plus algebra, etc., realized that we were speaking not in prose, but “in DEDS”. In some sense, a new multidisciplinary community, mostly fed by specialties as AC, OR and CS, but also from many others like Artificial Intelligence or several applications domains, started to be recognized.

Nearly a quarter of century ago, in the report about Future directions in control theory: a mathematical perspective, it was stated that (Fleming, 1988):

 [...] there exist no formalisms for DEDS mathematically as compact or computationally as tractable as the differential equations are for continuous systems, particularly with the goal of control.

Certainly the field is much more mature today, as it can be readily verified by looking at the thousands of published works and their applications to real problems. Roughly speaking, it can be said that the same basic operational formalisms remain today, PNs in particular, and considerable diversity still prevails in the DEDS arena.

The purpose of this invited contribution is not to provide a broad view of Petri’s life and contributions, neither to overview the development of the entire field. In Sec. 2 we will concentrate on providing just a few hints on the first question. Next we deal with a sketch of our modeling paradigm rather than a collection of “unrelated” formalisms (Sec. 4). Finally, in Sec. 5 we will briefly refer some of the interactions we had with Carl Adam over twenty-five years.

2. ABOUT CARL ADAM PETRI

Carl Adam Petri is born in Leipzig (1926). His father, professor of mathematics, enjoys friendly relationships with insigne mathematicians as David Hilbert and Hermann Minkowski. Obliged to join the German army (1944), he becomes prisoner in England (till 1946), where later he teaches Latin and chemistry. Petri returns to Germany in 1949, studying mathematics at the Technical University of Hannover (till 1956); from 1959 until 1962 he works at the University of Bonn; in 1962 he gets the doctorate degree from the Technical University of Darmstadt, obtaining an award for best doctoral thesis of the year in the Faculty of Mathematics and Physics. This singular work does not try to solve a more or less concrete problem. Petri deals with a number of fresh views, proposals to act, and insights. Later, until 1968 Petri is working at the University of Bonn, becoming head of the computer installations. In that year it is founded the GMD (Gesellschaft fur Mathematik und Datenverarbeitung), a national research center for applied mathematics and computer science. Carl Adam is called to manage the Institut fur Informationssystemforschung (Institute for Information Systems Research), being later one of the directors of the Institut fur Methodische Grundlagen (Institute for Methodological Foundations). In his prominent group work researchers as H. Genrich, K. Lautenbach, C. Fernández or K. Voss. He retires from GMD in 1991, but in 1988 was named Honorary Professor at University of Hamburg. In a village close to Bonn, Carl Adam died in 2010.

We can say that Petri was a mathematician in a computer environment. Nevertheless he was not a classical mathematician. Interested in the description of some real-life situations, he essentially worked at conceptual level, “opening windows” (i.e., providing foundations for new ways of thinking) to represent systems. This may be viewed as a profile less frequently exercised with success, than that of “theorem prover”. In other words, he was much more conceptual than technical. Moreover, when in Computer Science the paradigm was local computations on mathematically intricate problems, Carl Adam Petri looked for a Systems Theory, beyond what the problems in Informatics were. He did not merely do an extension of entities and constructions dealing with sequential computations. Certainly, he introduced a new and fresh approach to the conceptualization of concurrent and distributed systems, looking for a framework applicable to many types of concurrent and distributed discrete systems, doing that in a broad landscape of research fields (computer science, law, manufacturing, transportation, chemistry, epidemiology, demography,...). Example of this preoccupation, in Petri and Smith (2007) it is claimed:

Net theory has incorporated a touch of Pragmatics from its very beginning. It demands respect for e.g.: limitation of all resources; inherent imprecision of measurement; partial independence of actions and decisions; and existence of illusions (“discrete” and “continuous” models), as the core of its “pragmatic” attitude.

Against what it is frequently quoted, in his PhD dissertation the graphical notation on PNs does not appear. The well-known bipartite graphs with conditions/places (local state variables, the values called markings) and events/transitions (locally bounded actions to the neighbor places) will come some three years later. When the Academy of Transdisciplinary Learning and Advanced Studies awarded him with its Gold Medal, he confesses:

In what follows I will describe some of the less well-known features of my work. The graphical representation of structural knowledge which is now in widespread use I invented it in a playful mood in August 1939, and practised 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. In my dissertation on Communication with Automata, introducing the theory of such Nets in the context of Informatics, I did not mention my plaything. I did not want the theory to ap-
pear as a “graphical method” instead of a mathematical attack on the then prevailing Automata Theory, based on arguments taken from modern Physics. Only some years later, I was bold enough to propose Net Graphics as one of the standard features, and they were greatly welcomed.

On an axiomatic basis, Petri established an expressive formalism, able to model concurrency and synchronization straightforwardly, thus also cooperation (and competition). Replacing temporal order by causal order, he looked concurrency as mutually causal independent occurrences. Causal orders and preservation laws were inspired to Petri by the laws of Physics. With this point of view, PNs can reflect the structure of the system being modeled. Nevertheless, the ideas of Petri were in advance with respect to the needs at that moment. This is why we can understand a certain “crossing of the desert”, before PNs become well-accepted. In this sense, we should not forget the role played by the project MAC (initially for “Mathematics and Computation”, later backronymed, among other things, to “Multiple Access Computer”) of the MIT, particularly by Anatol W. Holt and his group, were participate researchers as M. Hack and F. Commoner, even A. Pnueli cooperate for a while. Holt “contributed substantially to the deepening and dissemination of net things”, as written by Carl Adam himself. Among many contributions, Holt directed the translation into English of the PhD dissertation of Petri, and named the nets as “Petri nets”. Thanks to Anatol Holt, the name of Petri is world well-known, and is constantly repeated.

In the 1970s, Robin Milner focusses his view of concurrent systems on the “interactions of smaller components” (Milner, 1980), one of the more celebrated steps in the definition of process algebras. In other words, Milner pays central attention to the process of construction of the model. Moreover, he also rejects the possibility of limiting the approach to computer systems, equally searching for minimality in the number of basic concepts. In this sense, in his Turing Award (1991), Robin Milner (1993) recognizes that several of these questions were already understood by Petri in the sixties, who [...] pioneered the scientific modeling of discrete concurrent systems. Petri’s work has a secure place at the root of concurrency theory. He declared the aim that his theory of nets should –at its lowest levels– serve impartially as a model of the physical world and in terms of its primitive constructions. What I always wanted to advance, to complement Petri net theory, is the synthetic or compositional view of systems which is familiar from programming.

The view proposed by Milner was algebraic, but this perspective can also be integrated in the PN theory, keeping what is sometimes called the “architecture” of the model; in other words, the trace of the way in which the model was constructed.

While bipartition at graph level was a salient feature in the Petri’s approach, it is not (explicitly) present in the one of Milner. Nevertheless, the use of bipartite graphs was not an exclusive feature in PNs, and other modeling approaches at the beginning of the sixties used it. For example, in queueing networks (QNs), were queues –as places– are static containers of clients, while servers in stations provide the services. Introduced as continuous systems, the Forrester Diagrams (Forrester, 1961), also meaningfully referred as stock and flow diagrams, use deposits or stocks as “warehouses”, the state-values being levels, while flows goes trough “valves”. It might be surprising, but Petri was not prolific in what generation of written works concerns: only some 35 works, several belonging to the so called “gray literature”. But the trace is impressive. It can be guessed in different ways.

For example, “The Petri Nets Bibliography”, that contains entries until 2006 only, has more than seven thousands, and it is uncertain how representative is this lower bound, that in reality may be even much less than 20% of the total. Another perspective comes from the conferences. Since 1980 it exists an annual conference devoted to PNs, the Int. Conf. on Application and Theory of Petri Nets and Concurrency (ICATPN; in fact, from 1980 until 1988 it was known as International Workshop, IWATPN); also from 1985 till 2003, ten editions of the IEEE/ACM Int. Symp. on Petri Nets and Performance Models (PNPM) (the first one was named: IEEE/ACM Int. Symp. on Timed Petri Nets) took place. This meeting merged in 2004 with others, as that on Process Algebra and Performance Models (PAPM), giving light to the annual Int. Conf. on Quantitative Evaluation of Systems (QEST). But what is more impressive is the number of meetings in which PNs is a topic explicitly mentioned, frequently devoting special sessions to them. For example, in “The Petri Nets: Meetings and Events” we can count more that 150 during 2005-2011, so more than twenty per year. Anyhow this would require some corrections, because meetings as WODES and ACC appear only once, while CDC, ECC and IFAC World Congress do not even appear! This may be partly understood because of the relative importance that is given to computer science/engineering conferences on that site with respect to automatic control and operations research. Additionally, let us point out the existence of many prestigious journals that devote to PNs at least one especial issue, and the existence of international standards and tools for their use in industrial environments.

During his career, Carl Adam Petri received several important recognitions. Among those, the Konrad-Zuse-Medal (1993); the Society for Design and Process Science established the Carl A. Petri Distinguished Technical Achievement Award (1997); the Werner-von-Siemens-Ring (1997), one of the highest ranked award for technical sciences in Germany, delivered every three years; the Doctorate

4 From the private correspondence of Petri to Holt; communication by Anastasia Pagnoni, at the Carl Adam Petri Memorial Symposium, the 4 of February, 2011, Berlin.
5 http://www.informatik.uni-hamburg.de/TGI/mitarbeiter/profs/petri_eng.html.
6 http://www.informatik.uni-hamburg.de/TGI/pnbib/.
7 Searching in Google Scholar “Petri net” and “Petri nets”, more than a quarter of million of entries are found, of course with redundancies. This number may be considered as an upper bound.
8 http://www.informatik.uni-hamburg.de/TGI/PetriNets/meetings/.
Fig. 1. The investiture as Doctor Honoris Causa by the Universidad de Zaragoza. José Manuel Colom and Manuel Silva were the sponsors.

Honoris Causa by the Universidad de Zaragoza, in 1999 (see also Sec.5); the Orde van de Nederlandse Leeuw (Order of the Dutch Lion), as Commander (2003); finally, in 2008, together with Edward J. McCluskey, he was awarded as IEEE Computer Pioneer: “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 and distributed computing”.

Meaningfully, most of the recognitions to Carl Adam Petri, a mathematician, came from engineering.

3. PETRI NETS: ON BASIC PROPERTIES AND WAYS OF APPROACH

Petri Nets are typical operational formalisms (i.e., they describe how the system works, not what it does). Among the qualities that they enjoy are:

Minimality in the number of primitives. This is usually a must in the construction of any conceptual framework, raising a classical trade-off between the engineering and scientific perspectives: while engineers appreciate a rich ontology with different concepts suited for different purposes, scientists mainly look for basic underlying notions. Obviously, the diversity and specificity of primitives may be convenient to develop concise and elegant models, but generally this tends to difficult formal reasoning and theory construction. An ideal solution to conciliate reasoning capabilities and practical expressivity consists on having a minimal number of basic primitives in terms of which richer ones can be constructed. In this sense, the basic PN formalism is quite spare, having only two simple and somehow orthogonal primitives (see in Fig. 2).

Locality and structuration. With PNs, the description is in local terms of states (marking of places) and state changes (transitions). Conducting in that way, it is possible to represent parallelism, synchronization, conflicts, etc. In fact, at a higher level, cooperation and competition relationships can be modeled. The locality of places and transitions is a central issue in PNs. It appears as a cornerstone for net models structuration. For instance, it is possible to refine a place or a transition (i.e., to detail the model), or to compose two modules by the identification of shared transitions (merging two or more into one) or places (fusion operation). Refinements and modularity can be based either on states or actions, thanks to their treatment on equal footing. In brief: both top-down and bottom-up modeling methodologies can be freely interleaved.

Temporal realism and occurrence nets: The representation of concurrency by means of causal independence of transition occurrences is more faithful than in the case of interleaving semantics (i.e., sequentialized views). This has several consequences. On one side, it allows to have “temporal realism” in timed interpretations, what is crucial for performance evaluation or control (scheduling...). From a different perspective, in untimed models, it allows to use unfolding techniques, what brings to occurrence nets, among other things, reducing the computational burden of purely sequentialized approaches (reachability or coverability graphs), something well understood, for example, for 1-bounded systems.

Although initially fully non-deterministic, PNs were provided with notions of time during the 1970s. QNs were initially defined in order to deal with resource contention among independent jobs. Thus, historically, the driving forces of QNs and PNs were different. An important limitation of initial proposals of QNs was the absence of a general construct to deal with synchronization. To overcome it, at the end of the 1970s, one of the focus was to provide high level primitives that supply the QN user with simple yet expressive building blocks. Informally, it can be “said” that PNs were developed in a bottom-up approach, while PNs (and process algebras) employed top-down approaches.

The classical proverb “all roads lead to Rome” is a clear recognition that Rome was a very (the most) important
town in Europe for several centuries. In analogous sense, given that using different roads we may arrive to “the town” of PNs (with several nuances not to be considered here), this may be “interpreted” as an indication of its relative importance.

To PNs we can arrive by different ways: (1) axiomatic (used by Petri himself); (2) by analogy with the state equation of continuous variable systems, what lead to vector addition systems; (3) by the theory of regions of a graph (like attributing particular codes to the states of an automata); or (4) by using the “subsequent calculi” of linear logic, among other possibilities. In the first case, we arrive to condition/event (C/E) systems, in the rest of cases to place/transition (P/T) systems. P/T systems are able to model infinite state (i.e., unbounded) systems, and constitute a “frontier formalism”. By this we means that it is very expressive and most of classical properties are decidable, but “small” extensions transform it in a formalism able to simulate Turing machines, thus excellent expressivity, but plenty of undecidabilities. Among the evoked extensions, just the addition of inhibitor (or zero test) arcs9 (alternatively, priority levels to the firing of transitions; or certain time interpretations). Abstraction to so called high-level PNs (Predicate/Transition, Colored, Object... PNs) is something like moving from assembler to high-level programming languages, or from a pure numerical to a symbolic level. Even if this last topic would require more explanations, it is here out of scope. Anyhow, from an engineering point of view, let us remark that in colored nets, for example, the model should express a good compromise between the “explicit” net structure (i.e., the relation of places and transitions) and the “implicit” structure (due to the functions associated to the arcs).

Of course, it is also out of scope to provide a bibliography on PNs. Even if certainly biased, I would like to leave some hints, most of them already historical. Up to our knowledge, following an initiative by Joseph Sifakis, Turin 1987, 2011 (see, for details: http://www.petrinets.info/standard.php). The final rapport was extensively diffused, for example, in Blanchard, M. et al. (1977) (also in other journals as: Automatisme, in Mars-April 1978). Here is worth remembering that, at the origin, the name came from “GRaphe de l’AF CET”, the learning society. The idea of diffusing it at an international level quickly “advised” to change the explanation of the acronym to “GRaphe Fonctionnel de Commande des Etapes-Transitions”. The interest of propagating it as an engineering norm bring, with the help of the ADEPA (“Agence nationale pour le Développement de la Production Automatisée”), to a French proposal in 1982: UTE NF C 03-190, Diagramme fonctionnel “GRAFCET” pour la description des systèmes logiques de commande. Years later, in 1988, appeared the IEC 848 standard, by the International Electrotechnical Commission (IEC): Preparation of function charts for control systems. It defines the Sequential Function Chart (SFC), a graphical programming language for PLCs based on Grafcet (see, later, the IEC 61131-3, of 1993).

If norms for logical automatisms were based on PNs through the Grafcet, it must be pointed out the existence of other normalization initiatives in complementary fields. For example, in the framework of software engineering, another important effort culminates with an International Standard Organization (ISO) norm. It is the ISO/IEC 15909, publicized into two parts: (1) Systems and software engineering - High-level Petri nets - Part 1: Concepts, definitions and graphical notation, 2004; and (2) Systems and software engineering - High-level Petri nets - Part 2: Transfer format, 2011 (see, for details: http://www.petrinets.info/standard.php).

4. PETRI NETS: A MODELING PARADIGM

The main argument in this section is that the conceptual seeds proposed by Carl Adam Petri do not just flourish on a single formalism (initially the so called Condition/Event nets, where the state is expressed in terms of boolean variables, thus only able to model finite state systems), but on a set of formalisms that constitute the foundations

---

9 In PNs the OR is present around places, by means of attributions and choices, while the AND is formed around transitions, by forks and joins. The inhibitor arcs “simply” add the boolean NOT.

10 About the society, see, for example, Hoffsaeb (1990).
of a modeling paradigm. The main idea in this section follow Silva and Teruel (1996).

A formalism is a conceptual framework that allows obtaining a kind of formal models of systems. For instance, ordinary differential equations are a formalism for the modeling of the dynamic behavior of continuous systems with lumped parameters. Some examples of formalisms for DEDS with diverse purposes are sequential state machines or state diagrams; Markov chains and different types of QNs (for performance evaluation); PERT graphs and conjunctive/disjunctive graphs (for scheduling); etc.

In view of the long life cycle of many systems (conception and modeling; analysis and synthesis from different perspectives; implementation and operation) and the diversity of application domains, it seems desirable to have a family of formalisms rather than a collection of “unrelated” or weakly related formalisms. Coherence among models usable on different phases, economy in the transformations and synergy in the development of models and theories are among the expected advantages. Following Kuhn’s ideas, a paradigm is (Kuhn, 1962):

 [...] the total pattern of perceiving, conceptualizing, acting, validating, and valuing associated with a particular image of reality that prevails in a science or a branch of science.

In particular, a modeling paradigm is a conceptual framework that allows to obtain formalisms from some common concepts and principles. Nevertheless, even if we believe that PNs constitute an adequate conceptual framework or paradigm for the operational description of DEDS, “we do not believe that it is always possible to select a single formalism, or family of them, to deal in a reasonable way with every aspect of every DEDS. The complexity and variety of systems suggest instead the interest of having multi-paradigm environments where the existence of sound and efficient bridges between different paradigms becomes a major issue” (Silva and Teruel, 1996). In a different, but close context, Milner (1993) states that:

I reject the idea that there can be a unique conceptual model, or one preferred formalism, for all aspects of something as large as concurrent computation.

In fact, even if it is natural trying to reduce the diversity of formalisms to a common framework (modeling paradigm), different languages, calculi and theories may be needed for the different possible perspectives.

In our view, the PN modeling paradigm derives from the “cross-product” of the different levels of abstraction on PNs (Condition/Event; Place/Transition; Colored...) and different interpreted extensions (for example considering external events or conditions and actions, adding time in one of the possible ways, etc) (see Fig. 3) As an example of synergy at the theory level, let us just mention that concepts as semiflows and invariants were immediately transferred from Place/Transition to Colored nets (even if at the computational level, nice and deep developments were needed); or that studying the computation of the visit ratio of transitions in a timed stochastic (Markovian) PN model, the so called rank theorems for the study of deadlock-freedom and liveness were found (Campos et al., 1991).11. The same kind of synergy can appear at the application level.

In multiformalisms approaches, there may be used unrelated formalisms, one per phase of the life-cycle. For example, automata (for functional specification), QNs (for performance evaluation), PERTs (for a basic control), different coding schemas (for the software implementation), etc. Eventually provided with adequate interpretations, PNs can do analogous kind of job: autonomous PNs, synchronized PNs or marking diagrams, stochastic PNs (many possibilities), etc., till the (semi)automatic generation of code, eventually fault-tolerant to a certain level.

Whereas local advantages in the modeling adequacy or in the specific solution techniques may appear using multiformalisms approaches, using a single paradigm like that of PNs, coherence among models of different phases, economies in the required transformations and analysis, and synergies can be expected. In any case, it is important to remark that, proceeding inside a given modeling paradigm does not mean at all that the work is not really multidisciplinary!

It is not possible to enter into the consideration of the use of formalisms of the PN-paradigm in the diverse applications domains. There exist too many! Let us point out their use for specification, design, verification, validation, simulation, implementation (eventually fault-tolerant), etc. in very different fields. Among others, in: logistics (external and internal), legal systems, asynchronous circuits, protocol engineering, software engineering, or workflow management. In most cases, methodological approaches are developed, and in several of those, software tools allow the practical use of concepts and methods. Last but not least, let us point out two complementary points of view in control engineering. They are both applicable to many domains and “classical”, in some sense. One is the so called Supervisory Control approach: given a plant and a logically desired (i.e., restricted) behavior, the design of a maximally permissive control is the goal. Different kinds of control problems, as computing the applicable actions in order to obtain the minimum time to reach a marking, are, in essence, Scheduling problems (like computing a 11 The results were generalized in a pure untimed framework (Teruel and Silva, 1996). Taking into account the construction process of the net (i.e., not viewing it as a flat structure), those kind of results were generalized to a class of multi-level nets (Recalde et al., 2001).
minimum makespan solution; i.e., problems in which the goal is to minimize the time difference between the start and the end of a sequence of jobs or tasks).

5. ON FIVE INTERACTIONS OVER TWENTY FIVE YEARS

In this section we briefly mention five points among our personal contact with Carl Adam. The first recalls our “disjoint” interests at the given time (1976). Later two topics, that we repeatedly discuss, are considered: synchrony theory and fluidization of net models. On both he shown strong interest. Even some potential cooperations were considered, but never substantiated.

The first time I met Carl Adam Petri was at the EN-SIMAG (Grenoble, 1976). My interest in meeting him was important because some time before I significatively changed the topic of my PhD work: from interconnected automata and modular hardware synthesis (with Cellules Universelles pour Séquences Ashynchrones, CUSAs), towards PNs and Programmable Logic Controllers (PLCs). At that time I was interested in software implementation issues (on PLCs and general purpose microcomputers), but he was considering problems of quite different nature (among others, related to synchronic distance, a metric in the firing of transitions). Our second selected point of interaction was around 1986, at the Bocconi University, where Anastasia Pagnoni organized an international meeting, and at the 7th European Workshop on Applications and Theory of Petri Nets (EWATPN), in Oxford. At that time I was interested in transitions firing dependencies in (weighted) PNs, partly working in cooperation with Tadao Murata. We discussed on synchronic distance (SD) (in chemistry, like an stoichiometric relation) and B-fairness (BF) relations. The SD relations deal with some kind of linear dependencies in the firing count of very long firing sequences; the BF relation is a generalization. It is a curiosity, Carl Adam appreciated the net system in Fig. 4, the “smallest” live, bounded and reversible ordinary net system having two transitions in BF relation, but not in SD relation (later, reproduced in Silva and Murata (1992)).

![Fig. 4. Transitions a and d are in B-fairness relation, but does not exist finite synchronic distance for them.](image)

The two following points I recall here happened at the University of Zaragoza. In 1987 we organize the 8th EWATPN and Carl Adam came to the workshop. With José Manuel Colom, we considered linear programming techniques (LP) in order to compute synchronic bounds and to characterize synchronic relations. Weak and strong duality and unboundedness theorems, and alternative theorems were considered for the first time in a systematic way (an improved version in (Silva and Colom, 1988)), what led to several polynomial time computations. Even if it was partly a computational topic, we discussed on it. Nevertheless, observe that this systematic use of LP has a relaxation of solutions into the non-negative reals implicit, what in some sense can be “interpreted” as dealing with autonomous continuous PNs (in fact, at the same meeting, continuous PNs were defined at the net level, not at the level of the state or fundamental equation, by our colleagues of Grenoble (David and Alla, 1987)). It should be said that continuous PNs were received with some reluctance in the EWATPN (later ICATPN) community, like stochastically timed PNs were some years before. The suspicious reception concerned in that time the evaluation of its relevance to the series of meetings, because not “truly discrete” models were being considered. Prepared as a tribute to Carl Adam Petri on the occasion of his 60th birthday, Concurrency and Nets (Voss et al., 1987) is a especial volume of “Advances in Petri Nets”. It was delivered to him at the banquet of the meeting, followed by “jotas aragonesas”, songs and dances from Aragon that Carl Adam did specially like.

In 1989 our school of engineering, Centro Politécnico Superior, celebrated its 25 years. We decided to make the first three Honoris Causa doctorates in Engineering by our university. For that purpose, in accordance with the three classical basic pillars of technology, three relevant personalities were selected: one from materials (Steve Tsai, Stanford University), another from energy (Amable Liñán, Universidad Politécnica de Madrid) and a third one from information: Carl Adam Petri, as representative of System Theory and Computer Science (GMD, Bonn). The day before of the main ceremony, we organized in his honor an international seminar with conferences of: G. Balbo (“On PNs and Performance Evaluation: The GSPN case”, Università di Torino); J. Billington (“On the ISO/IEC Petri Net standard (15909)”, University of South Australia); D. de Frutos (“Decidability properties in Timed Petri Nets”, Universidad Complutense de Madrid); M. Kontny (“Combining Petri Nets and Process Algebras”, University of Newcastle upon Tyne); and M. Silva (“On continuous Petri Nets”, Universidad de Zaragoza).

The last time I met him was in 2005, in Miami, at the 26th ICATPN. In that opportunity I had the honor to give a keynote on “Continuization of Timed Petri Nets: From Performance Evaluation to Observation and Control”. Therefore, we (re)discussed on fluid “views” of discrete models. Once again, he told me about the importance of the duality of reactants and reactions, what in time suggested the idea of the separation of places from transitions to him. In an enjoyable conversation I claimed that my game was “in moles, not in molecules, so I was able to consider fractions of moles”. Affectionately, he stated that my interest for fluid net systems was “not surprising”, because of my degree on Industrial-Chemical Engineering from the Universidad de Sevilla.

Carl Adam Petri always was a warm-hearted person.

---

12 Among the precedent works, H. Genrich and K. Lautenbach use LPP for computing in marked graphs, while G. Memmi and J. Sifakis use the so called Minkowski-Farkas’ lemma or alternative theorems in isolation.