Discovering Petri Nets
A Personal Journey

Wil M.P. van der Aalst

Eindhoven University of Technology, PO Box 513, Eindhoven, The Netherlands

1 Introduction

Carl Adam Petri (12 July 1926 – 2 July 2010) pioneered the computer science discipline and is one of the founding fathers of the wonderful field of concurrency. He single-handedly started a new subfield of computer science focusing on *concurrency* [11]. As Robin Milner phrased it in the acceptance speech for his Turing Award in 1991: "Much of what I have been saying was already well understood in the sixties by Carl Adam Petri, who pioneered the scientific modeling of discrete concurrent systems. Petri's work has a secure place at the root of concurrency theory!". Petri nets have become a standard tool to model and analyze processes where concurrency plays a prominent role. The ideas have been embedded in many other process modeling notations. For example, the widely used BPMN (Business Process Model and Notation) models use token-based semantics [9]. After working with Petri nets for over 30 years, I remain surprised by the elegance of the formalism. Using a few basic concepts (places, transitions, arcs, and tokens) and the simple firing rule, one enters a new "world" where it is possible to model a wide range of behaviors and study non-trivial phenomena (conflict, concurrency, confusion, etc.).

According to [12], Petri invented Petri nets in 1939 at the age of 13 for the purpose of modeling chemical processes. However, many refer to his PhD thesis defended in 1962 [10] as the starting point for Petri nets. This is only partially true, since his PhD thesis does not show the characteristic diagrams we know today. These emerged in the mid 1960-ties and subsequently conquered the world. Petri nets are used in a wide range of domains, directly benefiting from the theoretical foundations developed over the last 50 years.

In the remainder, I will describe how I got submerged into Petri nets and how it has affected my research and scientific career. Moreover, I will link two of Petri's guiding principles to my current research area:

– **GP1**: *Concurrency should be a starting point for system design and analysis and not added as an afterthought (locality of actions).*
– **GP2**: *A formalism should be consistent with the laws of physics and not take any shortcuts at the foundational level.*

Anyone familiar with Petri's work and his lectures will recognize these guiding principles. I will relate these two principles to process mining and the broader field of Business Process Management (BPM). Process mining can be viewed as the missing link between model-based process analysis and data-oriented analysis techniques [4].

Many process mining techniques use Petri nets or related formalisms and directly benefit from the above two principles proposed by Carl Adam Petri. This paper concludes by discussing Petri's heritage.

2 Personal Journey into the Wonderful World of Petri Nets

I met Carl Adam Petri for the first time in 1989 in Bonn while attending the 10th International Conference on Applications and Theory of Petri Nets. I was doing my PhD at the time and it was a very exciting and inspiring experience. This was my first Petri net conference, not knowing that many Petri net conferences would follow. Over the last 30 years, I have served as a program committee chair (twice, in 2003 and 2017), organized the conference once, served as a steering committee member since 2003, and played many other roles (e.g., chairing different workshops and committees).

For my Master project [1], I worked in the research group of Kees van Hee on a language and tool called *ExSpect*. In a way we "rediscovered" Petri nets, not knowing the seminal work done by Petri. Initially, we used triangles for places rather than circles and there were also other difference. However, the similarities between our work and (colored) Petri nets were striking. Therefore, we soon joined the Petri net community. I was responsible for adding time to our high-level *ExSpect* nets. *ExSpect* was developed concurrently with *Design/CPN* (that later evolved into *CPN Tools*, still hosted by our research group in Eindhoven). Both languages used colored tokens and provided hierarchy notions. During my PhD, I continued to work on *ExSpect*. My main focus was on the analysis of temporal behavior using both model checking and simulation [2]. The primary application domain for my work was the broader field of logistics. We analyzed supply chains, distribution centra, railway transport, container terminals, etc. using *ExSpect*.

After a few years I became the leader of a small research group in Eindhoven and my interests shifted to workflow management [5]. I noted the huge potential for applying Petri nets in this upcoming domain. We developed a framework for modeling and analyzing workflow processes based on Petri nets. This led to seminal notions such as WorkFlow nets (WF-nets) and soundness. Interestingly, we were able to use Petri nets notions such as invariants, siphons, traps, reduction rules, etc. to verify the (in)correctness of workflow models stored in commercial systems like *Staffware*.

After designing several workflow languages and developing several workflow management systems (YAWL, Declare, etc.), I got more and more interested in the *relationship between models and reality*. This was fueled by the repeated observation that simulation models (modeled in *CPN Tools* or *ExSpect*) rarely behave like real organizations, machines, and people. At the same time, workflow management research shifted from automation to *Business Process Management* (BPM). See [3] for a survey describing this transition. The scope of BPM extends far beyond workflow automation, including the understanding why processes and organizations achieve particular performance levels (time, quality, costs, compliance, etc.). Process improvement requires a deep understanding of processes that cannot be obtained through modeling only.

The desire to link models and reality naturally evolved into the new field of *process mining* around the turn of the century. We developed the first process mining algorithms around 1999. Initially, we used the term "workflow mining" rather than "process mining". Process mining starts from event data and uses process models in various ways, e.g., process models are discovered from event data, serve as reference models, or are used to project bottlenecks on. Many process mining techniques use Petri nets for obvious reasons. Later, I will elaborate on the role of Petri nets and Petri's guiding principles in process mining.

Although I worked on very different topics (logistics, simulation, workflow verification, workflow automation, BPM, and process mining), all of my research was (and still is) related to Petri nets in some form.

Fig. 1. Carl Adam Petri (middle) and Grzegorz Rozenberg (left) during the Petri net conference we organized in 2003. The photo was taken just after Petri was honored with the prestigious title "Commander in the Order of the Netherlands Lion" by Her Majesty the Queen of the Netherlands.

Concurrently, I moved up the academic ranks and became assistant professor (1992), associated professor (1996), and full professor (2000). In 2003, I organized the Petri net conference in Eindhoven together with Kees van Hee. This conference was a big success and quite special because Carl Adam Petri gave a talk after not having attended the conference for many years. Grzegorz Rozenberg (who can be considered as the "godfather" of our wonderful Petri net community) encouraged me to organize a co-located event next to the Petri net conference. This resulted in the first Business Process Management conference (BPM 2003). Also within the BPM community Petri nets were adopted as a standard tool to model, analyze, and enact business processes.

BPM is just one of many fields using Petri nets, illustrating the foundational nature of Petri's ideas.

3 Concurrency is Key

Petri's first guiding principle is "Concurrency should be a starting point for system design and analysis and not added as an afterthought (locality of actions)" (**GP1**). Many other modeling approaches start from a sequential view on the world and then add special operators to introduce concurrency and parallelism. Petri nets are inherently concurrent. Although Petri nets are often seen as a procedural language, they can be viewed as declarative. A Petri net without any places and a set of transitions T allows for any behavior involving the activities represented by T. Adding a place is like introducing a constraint. The idea that transitions (modeling activities or actions) are independent (i.e., concurrent) *unless specified otherwise* is foundational! This allows us to model things in a natural manner and also facilitates analysis. Actions are local and this allows us to understand things better while enabling "divide and conquer" approaches (e.g., decomposing analysis problems).

Mainstream notations for modeling processes use token-based semantics adopted from Petri nets. The de facto standard for business process modeling—BPMN (Business Process Model and Notation) [9]—uses token passing. Also UML activity diagrams use token-based semantics and a notation similar to Petri nets. Unfortunately, these languages provide a plethora of control-flow constructs basically killing the elegance of the original proposition. However, in the back-end of such languages and supporting systems, one can often find Petri nets. For example, BPMN models are often translated to classical Petri nets for verification.

4 Process Mining: Relating Observed and Modeled Behavior

Petri's second guiding principle is "A formalism should be consistent with the laws of physics and not take any shortcuts at the foundational level" (**GP2**). He often related concurrency theory to physics [7, 13]. However, the principle can also be applied to everyday's discrete event processes (e.g., in manufacturing, healthcare logistics, luggage handling systems, software analysis, smart maintenance, website analytics, and customer journey analysis). We seek models adequately describing these real-world phenomena. Interestingly, the digital universe and the physical universe are becoming more and more aligned making it possible to study these discrete event processes much better. The spectacular growth of the digital universe, summarized by the overhyped term "Big Data", makes it possible to record, derive, and analyze *events*. Events may take place inside a machine (e.g., an X-ray machine, an ATM, or baggage handling system), inside an enterprise information system (e.g., an order placed by a customer or the submission of a tax declaration), inside a hospital (e.g., the analysis of a blood sample), inside a social network (e.g., exchanging e-mails or twitter messages), inside a transportation system (e.g., checking in, buying a ticket, or passing through a toll booth), etc. [4]. Events may be "life events", "machine events", or "organization events". Earlier, I coined the term *Internet of Events* (IoE) to refer to all event data available [4].

The event data that are abundantly available allow us to relate real-life behavior to modeled behavior. More specifically, we can learn process models from such event data (process discovery) or replay event data on models to see discrepancies (conformance checking). This is exactly what process mining aims to do.

Process mining starts from *event logs*. An event log contains event data related to a particular process. Each event in an event log refers to one process instance, often called case. Events related to a case are ordered. Events can have attributes. Examples of typical attribute names are activity, time, costs, and resource. *Process discovery* is one of the most challenging process mining tasks. Based on an event log, a process model is constructed thus capturing the behavior seen in the log. Dozens of process discovery algorithms are available and many produce Petri nets. Input for *conformance checking* is a process model having executable semantics and an event log. Discrepancies between the log and the model can be detected and quantified by replaying the events in the log. Simple conformance checking approaches try to play the token game and count missing and remaining tokens. More sophisticated approaches solve optimization problems to find modeled behavior most related to the observed behavior. Some of the discrepancies found may expose undesirable deviations, e.g., conformance checking signals the need for better controls. Other discrepancies may reveal desirable deviations and can be exploited to improve process support.

The empirical nature of process mining immediately exposes formalisms not being able to capture real-life behavior. Choosing the wrong "representational bias" results in discovered models that are poorly fitting (observed behavior is not allowed or the model is over-fitting or under-fitting) [4].

Petri nets are attractive for process mining given the abundance of analysis techniques. For example, conformance checking techniques use the marking equation to dramatically reduce the search space when computing alignments. Moreover, the fact that "a Petri net without any places and a set of transitions T allows for any behavior involving the activities represented by T" is a great starting point for process discovery. Obviously, such a Petri net is underfitting, but additional constraints can be introduced by adding places. This is related to the seminal idea of *regions* (both language-based regions and state-based regions) [6, 8]. The synthesis of Petri nets based on regions is one of the cornerstones of process discovery, very much in the spirit of Petri's second guiding principle.

5 Petri's Heritage

This short paper focused on two of Petri's guiding principles: (1) concurrency should be a starting point for system design and analysis (and not added as an afterthought), and (2) a formalism should be consistent with the laws of physics and not take any shortcuts at the foundational level. I linked these two principles to my own research over the last 30 years and discussed how these principles relate to the emerging field of process mining. Obviously, concurrency of behavior and consistency with reality are key notions in process mining. However, above all, this paper described a personal journey reflecting on the influence of Petri's work on my career and research aimed at discovering Petri nets from events.

Carl Adam Petri discovered his nets at a time that information processing was viewed as something sequential. Formal notations for concurrency and asynchronous distributed systems were uncovered by Petri's seminal work. Petri nets are used in many domains and the strong theoretical foundation often helps to solve "wicked problems" and avoid reinventing the "wheels of concurrency". For example, numerous workflow management, BPM, and process mining approaches directly build on Petri nets.

However, it remains crucial to invest in the foundations of non-sequential processes. Einar Smith's book on Petri's life and achievements [13] provides interesting insights into the "good old days" of scientific research at the Gesellschaft für Mathematik und Datenverarbeitung (GMD). At GMD in Schloss Birlinghoven there was still time to work on the theoretical foundations of computing. This is in stark contrast with today's research practices driven by "quick wins" and application-oriented projects rather than long-term scientific goals. Today's scientists simply do not have the time to take a step back and ask long-term questions in the way Carl Adam Petri did. *Would Petri have survived today's research environment?* As part of his heritage we should ask ourselves this question repeatedly. This may help us to create better research environments working on the true foundations of computing.

References

1. W.M.P. van der Aalst. Specificatie en Simulatie met behulp van ExSpect (in Dutch). Master's thesis, Eindhoven University of Technology, Eindhoven, 1988.
2. W.M.P. van der Aalst. *Timed coloured Petri nets and their application to logistics*. PhD thesis, Eindhoven University of Technology, Eindhoven, 1992.
3. W.M.P. van der Aalst. Business Process Management: A Comprehensive Survey. *ISRN Software Engineering*, pages 1–37, 2013. doi:10.1155/2013/507984.
4. W.M.P. van der Aalst. *Process Mining: Data Science in Action*. Springer-Verlag, Berlin, 2016.
5. W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods, and Systems*. MIT press, Cambridge, MA, 2002.
6. E. Badouel, L. Bernardinello, and P. Darondeau. *Petri Net Synthesis*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2015.
7. W. Brauer and W. Reisig. Carl Adam Petri and Petri Nets. *Informatik-Spektrum*, 29(5):369–374, 1996.
8. A. Ehrenfeucht and G. Rozenberg. Partial (Set) 2-Structures - Part 1 and Part 2. *Acta Informatica*, 27(4):315–368, 1989.
9. OMG. Business Process Model and Notation (BPMN). Object Management Group, formal/2011-01-03, 2011.
10. C.A. Petri. *Kommunikation mit Automaten*. PhD thesis, Fakultät für Mathematik und Physik, Technische Hochschule Darmstadt, Darmstadt, Germany, 1962.
11. C.A. Petri. Introduction to general net theory. In W. Brauer, editor, *Net theory and applications : Proceedings of the advanced course on general net theory,processes and systems (Hamburg, 1979)*, volume 84 of *Lecture Notes in Computer Science*, pages 1–20. Springer-Verlag, Berlin, 1980.

12. C.A. Petri and W. Reisig. Petri Net. *Scholarpedia*, 3(4):6477, 2008. http://www.scholarpedia.org/article/Petri_net.
13. E. Smith. *Carl Adam Petri: Life and Science*. Springer-Verlag, Berlin, 2015.