

# Carl Adam Petri and "Petri Nets"

## Wilfried Brauer, Wolfgang Reisig

Scientific tradition frequently names a fundamental notion, insight, or theory after its explorer or eminent representative. Typical examples include "*Abelian group*", "*Planck's constant*" or "*Keynesian economic theory*". Informatics occasionally names algorithms after their inventor, such as Dijkstra, Lamport or Floyd. Entire subfields of informatics are named after persons only rarely. One of these few persons is *Carl Adam Petri*. Probably every professional knows "*Petri Nets*" as a modeling technique. This paper will survey Petri's exceptional life and work.

Petri started his scientific career with his dissertation "Communication with Automata", which he submitted to the Science Faculty of Darmstadt Technical University in July, 1961. He defended his thesis there in June, 1962. [1]

In the rest of this paper we first discuss this unusual and exceptional work, which lay the foundations for the success story of PetriNets including stimulations and challenges still to be taken up. We then consider the personal background of Carl Adam Petri and the prerequisites and motivation for his work. We finish with the impact of Petri's dissertation and his other publications onto the evolution of informatics.

All together, it will become transparent why some of Petri's ideas from the early 1960ies had been taken up much later, and why some of his ideas are still pending for discovery.

### **The dissertation**

Since its publication in 1962, Petri's dissertation has been quoted frequently (even though it probably has been read much less frequently).

This is not a conventional PhD dissertation solving an open problem or elaborating a new theory. Instead, like in many later papers, Petri presented a wealth of ideas and proposals for revising foundations of informatics. The text thus resembles sketches of a research program to some degree.

Nevertheless, as an excellent motivation for this fundamentally new approach, Petri starts out with an absolutely concrete and practical problem concerning the computation of recursive functions. The problem focuses on the observation, which was well known in those years already, that for a general recursive function  $f$  and an argument  $n$ , the amount of intermediate space necessary to compute  $f(n)$  can not be assessed in advance. Consequently, you can not get hold of the required resources and then compute  $f(n)$ . Instead, you have to start out with a given set of resources. If the resources suffice, meaning that the computation of  $f(n)$  terminates, you had good luck. Otherwise, you have to assemble more resources and start again. Petri challenged the necessity of re-starting from scratch: can't you organize a computing system in a way that fresh components can be allocated whenever necessary and that the computation continues right away after adding components? Of course, the number of extensions must be unlimited and extensions shouldn't significantly slow down the computation. Conventional computer architectures fail, as the following arguments show. Each extension enlarges the system's overall size. This requires longer wires, in particular to the clock pulse. This, however, lengthens the runtime of the signals. Hence, the clock frequency must be reduced. Furthermore, the clock pulse generator's fan-out increases and consequently its power consumption increases without limits. Clock frequency and power consumption of a switching element can not be changed at will. The clock pulse generator eventually collapses.

Therefore, the question arises whether there exists an extendable architecture that does not need longer wires and which does not suffer from growing fan-out. Petri has proven that this can be achieved: attach each fresh component to that component having been attached previously. This

construction comes with a price: each component must be able to act autonomously and the entire system works asynchronously.

As a feasibility study, Petri designed an asynchronous pushdown device: a sequence of *modules*, with each module containing a single data element and communicating with its two neighbors. The most recently attached module has one neighbor only. This way, a further fresh module can be attached. It is well known that two such devices suffice to implement a Turing Machine. Hence, Petri's construction is computationally universal! Petri presented this architecture at the first IFIP World Computer Conference in Munich, 1962. [2]

With the help of this thought experiment, Petri intended to show that asynchronous systems are more powerful than synchronous systems. From this insight he deduced the consequence that a general theory of information processing, if intended to be practically relevant and not unnecessarily idealizing, *must* start out with asynchronous, locally limited operations. It is therefore inappropriate to base the theory of informatics on sequential models.

In the course of his work, Petri employs a multitude of formal notations for asynchronous, distributed systems, including graphical representations and algebraic formulae with a "parallel" operator, in analogy to what later became process algebra. He additionally coined the basic notions of Petri Nets, i.e. "places" to describe local states and "transitions" for locally bounded actions.

Petri Nets – as they are known these days – first appeared in Petri's 1965 talk "Fundamentals on the description of discrete processes" at the 3<sup>rd</sup> Colloquium on Automata Theory in Hannover, 1966 [3]. However, at the end of 1964 already, the well known software pioneer Tom DeMarco got in touch with Petri Nets at Bell Telephone Laboratories in the ESS-4 project (developing "the world's first commercial stored program telephone switch"). Tom has been a member in the project's simulation team. In his contribution to the volume on "Software Pioneers" [4], DeMarco writes "Among the documents describing the simulation was a giant diagram that Ms. Hoover called a Petri Net (Ms. Erna Hoover ran the team). It was the first time I had ever seen such a diagram. It portrayed the system being simulated as a network of sub-component nodes with information flows connecting the nodes. In a rather elegant trick, some of the more complicated nodes were themselves portrayed as Petri Nets..." And some lines later: "The one document that we found ourselves using most was Erna's Petri Net. It showed how all the pieces of the puzzle were related and how they were obliged to interact. The lower-level network gave us a useful pigeon-holing scheme for information from the subsystem specs. When all the elemental requirements from the spec had been slotted by node, it was relatively easy to begin implementation. One of my colleagues, Jut Kodner, observed that the diagram was a better spec than the spec".

Petri did not just demand an adequate modeling technique for asynchronous distributed systems. His technique should additionally meet a number of further requirements.

First of all, his modeling technique should obey the laws of physics. In particular, this implies to give up the fiction of global states. A discrete action of a system usually does not affect all system components, but only a few of them. An evident example is a computing step of an internet embedded pc: it is not adequate to conceive such a step as an update of the internet's global state. Petri suggested to model the *locality* of actions during system runs with utmost precision and to respect and exploit this aspect. Describing an action as a pair of an old state and a new state, as it is usually done, represents this aspect only implicitly. It is entirely inadequate to represent a single system run as a sequence of (global state occurrences and) action occurrences. Petri suggests action occurrences not to be ordered along a fictive, idealized time scale, but by the partial order induced by the cause and effect relation instead. Two action occurrences *a* and *b* may remain un-ordered. This happens in case none of them depends on the outcome of the other. Un-order then represents causal independence ("concurrency", in Petri's terminology). Observe that *a* may be causally independent of *b*, and *b* may be causally independent of *c*, with *a* causally before *c*. Hence, concurrency is not necessarily a transitive

relation, in contrast to "temporally coincident". Relativity theory likewise assigns a "pre-cone" and a "post-cone" to each  $a$ . These cones consist of the action occurrences causally before and after  $a$  respectively. Many years later this was re-detected and re-formulated in other system models by Lamport, Pratt, Gurevich and others. Transferring the idea of causal order from physics to informatics is a particularly impressive example for Petri's demand to design the theory of informatics in accordance with the laws of physics.

As a second requirement, Petri suggests to form models of informatics in the tradition of models of science. Profound scientific theory is rooted in *preservation laws*. Examples are substance equations of chemistry or energy preservation laws of physics. Elementary discrete actions should likewise obey a law of preservation. A typical necessary criterion for preservation laws is the *reversibility* of processes. In informatics this means for a step  $S \xrightarrow{a} S'$  of an action  $a$  from a state  $S$  to a state  $S'$ , that not only  $S'$  can be computed from  $S$  and  $a$ , but that  $S$  can be re-computed from  $S'$  and  $a$  as well. As an example, the assignment statement  $x := x + 1$  is reversible, whereas  $x := 1$  is not. Petri Net transitions are designed such that *local* causes and effects become evident and that they are reversible.

Petri's third requirement demands that a modeling technique should not only be adequate to describe implementable behavior, but it should also describe men's *pragmatic use* of computing systems. This aspect motivates the dissertation's ambiguous title "Communication with Automata": it covers men communicating with automata, as well as the communication between different persons *with the help of* automata. Petri outlined these aims, together with ideas how to reach them.

This has not been a conventional dissertation – it was more to be conceived as a program of how to lay the foundations for the emerging science of informatics. This program however, was disconcerting in its long term aims and contradicted prevailing ideas in its short term proposals. So it is easy to imagine that this piece of work caused its readers quite a headache. One of the leading pioneers of the first generation electronic computers in Germany, Prof. Alwin Walter, of Darmstadt Technical University, recognized the value of Petri's work and ensured that it was awarded as the best dissertation of his school in 1961/1962.

Petri's dissertation has been translated to English: in the context of the venture MAC at MIT, in Anatol W. Holt's "Information Systems Theory Project" [5].

## **Carl Adam Petri**

Distributed and concurrent processes as the foundations of informatics, theoretical constructs in accordance with physical laws, preservation laws in analogy to science, formal modeling of the pragmatic use of computing systems – exotic topics for the emerging science of informatics in the early 1960ies, and rarely expected in a dissertation. Who is the person to build up his own scientific world, detached from actual fashion and tendencies, but nevertheless respecting practical needs and scientific tradition? Petri's biographical data are typical for his age-group and quickly reported. Decisive experiences as a child and young man influenced his scientific work later on.

Carl Adam Petri was born in Leipzig in 1926. He graduated in 1944 at the famous Thomas school and was immediately forced into military. He was a British prisoner and remained in England until 1949. He then studied mathematics in Hannover and followed his teacher Heinz Unger as a PhD student to Bonn University. After getting his PhD in 1962 he set up and ran the computer center of Bonn University, as well as the Institute for Information Systems Research at the "Gesellschaft fuer Mathematik und Datenverarbeitung" (GMD) in Birlinghoven (meanwhile an institute of Fraunhofer society). He ran that institute until 1991. He turned down an offer for a position as a full professor at the University of Dortmund in 1973. During his entire career as well as after his retirement, Petri developed and published those ideas already sketched in his dissertation.

A number of reports and anecdotes help explain why Petri focused on very special scientific problems and methodological approaches. We mention three of them:

Petri's father had a PhD in mathematics and still had met Minkowski and Hilbert. He supported his son's interest in science. From a bookseller's bankrupt estate, Petri got two thick textbooks on chemistry at his 12<sup>th</sup> birthday, which he worked through. His father organized him an exceptional permit to use the Leipzig central library unrestrictedly. There he delved into publications of Einstein and Heisenberg.

Young Carl Adam as a flak auxiliary at the air force observed officers, who estimated the height, distance and speed of approaching aircraft by simple means including visual judgment and hearing. The combination of measurement and estimation and mainly the quest for the responsibility for – inevitable – mistakes, has occupied him and influenced much of his scientific work.

In his years in England, Petri has solved some challenging land measurement problems, such as the construction of concentric ellipses on rolling country.

The three topics mentioned, i.e. the methods of science, the pragmatic aspects of erroneous measurement, and the arguments on geometrical objects, turn up again in Petri's later scientific work.

### ***The years until 1980***

Petri's dissertation has initially rarely been taken notice of. This is hardly surprising, considering its environment: computing technology at that time consisted of huge mainframe computers for numerical calculation, with punched paper tape and punch cards for input and output. A global clock seemed to be a natural course of action. Nobody thought about computer networks. Application software focused on numerical problems. Since 1956 such software was mainly written in FORTRAN. The ALGOL 60 language initiated systematic research on the theory of sequential programs and sequential processes. Petri's proposal to organize input and output as parallel processes had been rejected. Missing I/O standards would hamper the success of ALGOL 60 later on. Applied informatics favored sequential processes. Mastering analog computers and asynchronous switching networks appeared to be difficult and too slow, despite its parallel execution.

Theoretical informatics focused on sequential models for automata and computers, as well as on computable foundations, formal languages and compiler theory. In addition, most speculative theory papers on "cybernetics" and "artificial intelligence" circulated, mostly without any realistic ideas on their realization (for example, speculations on a "general problem solver"). In contrast, Petri outlined the feasibility of his proposals by using the technology of that time. Even more, he discussed pragmatically the adequate use of this technology.

Summing-up: Petri's proposals were too early for applied informatics. And theory focused on other topics. However, the poor response did not confuse Petri. Together with his staff and numerous visitors he strengthened his proposals in the 1960ies and 1970ies.

Still in the 1960ies he formulated a couple of basic principles that have been re-invented by others later on. This includes alternating local states and steps (later in message-sequence-charts), side effect free actions (later in functional languages) and multisets of tuples as states together with "put" and "take" replacing "write" and "read" (later in the "tuple space" of LINDA and in the Chemical Abstract Machine).

On the occasion of his acceptance speech for the Turing Award 1991, Robin Milner set out "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."

As a first breakthrough, the MIT considered Petri Nets in the project MAC, at the end of the 1960ies. This successful project on "Multiple Access Computer" (and other topics) reflected some of Petri's ideas. This gave rise to a line of research conceiving Petri Nets as a mechanism to characterize classes of Petri Nets with the help of formal languages. Some insight into Petri Nets can be gained this way indeed. But Petri was reluctant to appreciate this kind of results: they confuse concurrent and causally ordered event occurrences, whereas Petri considers this distinction to be fundamental in order to conceive systems properly. A number of theoretical results on Petri Nets emerged in the 1970ies, in particular linear-algebraic methods to compute system invariants.

Essentially all theoretical results and applied case studies, available in the late 1970ies, have been compiled and presented at the First Advanced Course on Petri Nets in Hamburg, 1979 (published as LNCS 84 [6]). In those years, distributed systems gained more attention and alternative modeling- and analysis techniques have been suggested, including process algebras and temporal logic.

### **The years since 1980**

The number of publications on Petri Nets grew sharply since the early 1980ies. As an entirely new concept, the marking of places with sets of uniform "black" tokens has been generalized to individual, "colored" tokens of different kinds. This step increased the modeling power of Petri Nets decisively, while structurally retaining the fundamental analysis technique, in particular the linear algebraic calculi of place invariants and transition invariants.

The 1980ies saw a growing interest in distributed and reactive systems, boosting not only Petri nets, but a number of alternatives alike. Many of them vanished without trace. The durable ones include the already mentioned process algebras with Robin Milner's  $\pi$ -calculus of the 1990ies, as well as David Harel's Statecharts. Starting with Amir Pnueli's seminal work in the late 1970ies, temporal logic became the favorite analysis technique of the field.

There is no clear evidence to what extent the graphical form of Petri Nets may have influenced the graphical form of other calculi. The above described idea of conceiving a distributed run as a partially ordered set of action occurrences, has initiated *partial order model checking* conventional model checking cannot cope with partially ordered runs, but only with total orders. A partial order, however, corresponds to a *set* of total orders. Some temporal logic properties hold for either all or none of those total orders in each set. So, it can effectively be tested for all total orders in a set by testing just one of them.

Petri Nets nowadays contribute to the actual discussion of model driven software design. Many actually favored modeling techniques include Petri Net based components; most prominently the activity diagrams of UML 2.

Meanwhile, Petri Nets are well established as a technique to model and to analyze computer embedded systems. A large community of scientists and software engineers employ Petri Nets in fairly different projects. Less known to computer scientists is the mechanical engineers' high esteem for Petri Nets. Carl Adam Petri has been awarded the 30<sup>th</sup> Werner-von-Siemens-Ring as a "scientist and designer of technology of outstanding merit". Petri is the second computer scientist who got honored with this prize (the first one was Konrad Zuse). In his laudatory speech, Prof. Gottzein pointed out: "Petri Nets brought engineers a breakthrough in their treatment of discretely controlled systems. Petri Nets are a key to solve the design problem, as this is the first technique to allow for a unique description, as well as powerful analysis of discrete control systems. Based on Petri Nets, it is now possible to formulate system invariants for discrete systems".

Nowadays, a number of useful software tools are available, mutually linked with the help of the "Petri Net Kernel", to design and to analyze Petri Nets. Conferences are organized on a regular basis, in particular the annual "International Conference on Applications and Theory of Petri

Nets", with satellite tutorials, workshops etc., since 1980. Meanwhile, the special interest group "Petri Nets and related system models" publishes a regular newsletter with almost 60 issues. The computer lab of the University of Hamburg organizes the internet portal [www.informatik.uni-hamburg.de/TGI/PetriNets/index.html](http://www.informatik.uni-hamburg.de/TGI/PetriNets/index.html) with a wealth of current references to literature, tools, events etc.

## Honors

With the establishment of Petri Nets as a recognized modeling technique in the 1980ies, Carl Adam Petri was honored many times. The outstanding awards include

- 1988 Verdienstkreuz 1. Klasse des Verdienstordens der Bundesrepublik Deutschland,
- 1988 Honorary professor at Hamburg University,
- 1989 Member of Academia Europaea, London,
- 1993 Konrad-Zuse-Medaille der Gesellschaft für Informatik, for distinguished credit for informatics,
- 1997 Werner-von-Siemens-Ring, for outstanding merits for techniques in connection with science,
- 1997 Member of the New York Academy of Sciences,
- 1999 Honorary doctorate of the University of Zaragossa,
- 2003 Commandeur in de Orde van de Nederlandse Leeuw.

## What will future bring?

In his invited speech at the 26<sup>th</sup> International Conference on Application and Theory of Petri Nets, Miami, June 2005, Petri appreciated the diversity and the quality of applications of his theory. But he called for a substantial expansion of the theory. Not for another bunch of Petri Net classes, or more sophisticated analysis algorithms, but for taking up the long term aims as outlined in his dissertation. Much remains to be explored! It may be a matter of time until progress in hardware or demands of software will be strong enough to recall Petri's old proposals.

An example for such a long term aim are the preservation theorems of information processing, in analogy to the preservation theorems of physics (such as Einstein's  $e = mc^2$ ) or chemistry (such as equations like  $\text{NaJ} + \text{Cl} \rightarrow \text{NaCl} + \text{J}$ ). Maybe – as Petri speculates – future software interfaces may be formulated much more abstract as well as more precisely, compared with today's state of the art. This, however, would require a fundamental notion of what is preserved during dynamic information processing. As a first approximation, Petri suggests a new notion of "information". Information *processing* then would mean to reshuffle the given information, while its *overall amount* remains constant. Apparently none of the presently used notion of "information" would meet this requirement. It is yet entirely unknown whether or not there is a corresponding smallest unit of information. Up until now Petri is still looking for such notions. Informatics evolved fiercely and unplanned, driven by its technological and economical potential. This kind of progression, together with today's common short time projects, affects the structure of scientific development. Petri rarely would have got a chance in this kind of environment. It is only a few, who ask long term questions on a systematic science of informatics like him.

## References

- [1] C. A. Petri                      Kommunikation mit Automaten. Schriften des Rheinisch-

Westfälischen Institutes für Instrumentelle Mathematik an der  
Universität Bonn Nr. 2, 1962

- [2] C. A. Petri            Fundamentals of a Theory of Asynchronous Information Flow. 1<sup>st</sup>  
IFIP World Computer Congress. Munich 1962, North Holland, pp  
386-390, 1963
- [3] C. A. Petri            Grundsätzliches zur Beschreibung diskreter Prozesse.3.  
Kolloquium über Automatentheorie, Birkhäuser-Verlag, pp121-  
140, 1967
- [4] M. Broy and            Software Pioneers. Springer-Verlag, 2002  
E. Denert (ed.)
- [5] C. A. Petri            Communication with Automata. New York: Griffiss Air Force  
Base, Technical Report, RADC TR-65-377-vol-1-suppl-1 Applied  
Data Research, Princeton, NJ, Contract AF 30(602)-3324, 1966
- [6] W. Brauer (ed.)        Net Theory and Applications. LNCS 84, Springer-Verlag, 1979

This paper has been translated from Informatik-Spektrum, Vol. 29, Nr. 5, pp 369-374, Springer-Verlag, 2006.