

**Erschienen in: Informatik Spektrum 29, Heft 5 Oktober 2006,
S.369-374**

Carl Adam Petri und die "Petrietze"

Einleitung

In der wissenschaftlichen Tradition werden in vielen Fächern zentrale Begriffe, Einsichten und Theorien mit den Namen ihrer Entdecker oder ihrer bedeutenden Vertreter belegt, beispielsweise "Abelsche Gruppen", "Plancksches Wirkungsquantum" oder "Keynes'sche Wirtschaftstheorie". In der Informatik ordnet man gelegentlich einzelne Algorithmen ihren Erfindern zu, beispielsweise Dijkstra, Lamport oder Floyd. Ganze Gebiete werden aber nur selten nach Personen benannt. Eine solche Person ist *Carl Adam Petri*; vermutlich jeder Informatiker kennt "*Petrietze*" als Modellierungstechnik.

Am 12. Juli ist Carl Adam Petri 80 Jahre alt geworden; ein Anlass, sein außergewöhnliches Leben und Werk ein wenig genauer anzusehen.

Petris wissenschaftliches Werk beginnt mit seiner Dissertation "Kommunikation mit Automaten", die er im Juli 1961 an der Fakultät für Mathematik und Physik der Technischen Universität Darmstadt eingereicht und im Juni 1962 verteidigt hat. Wir diskutieren zunächst diese in der Tat un- und außergewöhnliche Arbeit, die der Erfolgsgeschichte der Petrietze zu Grunde liegt und deren vielfältige Anregungen bis heute noch nicht alle aufgegriffen wurden. Danach fragen wir nach dem Hintergrund der Person Carl Adam Petri und nach den Voraussetzungen und Motiven für diese Arbeit. Wir enden mit Bemerkungen zur Wirkungsgeschichte von Petris Dissertation und seinen anderen Arbeiten.

Dabei wird deutlich, weshalb einige seiner Ideen aus den 60 Jahren erst sehr viel später aufgegriffen oder neu entdeckt wurden und dass einige seiner Vorschläge noch darauf warten, ausgearbeitet zu werden.

1. Die Dissertation

Seit ihrer Publikation 1962 wird Petris Dissertation oft zitiert (vermutlich aber deutlich weniger oft gelesen). Diese Arbeit ist keine konventionelle Dissertation, in der etwa ein offenes Problem gelöst oder eine neue Theorie vorgestellt und ausgearbeitet wird. Wie in vielen seiner späteren Arbeiten auch, formuliert Petri eine Fülle von Anregungen und Vorschlägen zu einer neuen Grundlegung der theoretischen Informatik, die eher Skizzen zu einem Forschungsprogramm ähneln.

Allerdings beginnt Petri die Arbeit mit einem ganz konkreten Problem der maschinellen Berechnung rekursiver Funktionen, das eine exzellente Motivation für seine grundlegenden neuen Vorschläge bildet. Dieses Problem beruht auf der (auch damals schon) bekannten Feststellung, dass man für eine rekursive Funktion f und ein Argument n im Allgemeinen nicht abschätzen kann, wieviel Speicherplatz die Berechnung von $f(n)$ auf einem Computer benötigt. Man kann also nicht zunächst die benötigten Ressourcen beschaffen und dann die Berechnung durchführen. Stattdessen muss man zunächst mit den vorhandenen Ressourcen beginnen. Wenn sie reichen, also die Berechnung terminiert, hat man Glück gehabt. Wenn nicht, muss man mit einem größeren Rechensystem von vorn beginnen. Petri fragt nun, ob man den wiederholten Neubeginn der Berechnung nicht vermeiden kann, indem man das zu kleine Rechensystem nur jeweils um einige Ressourcen ergänzt und damit weiter rechnet. Solche Erweiterungsschritte sollten nach beliebig vielen vorausgegangenen Erweiterungsschritten immer wieder möglich sein

und zwar sogar ohne die Berechnung ständig zu verlangsamen – jedenfalls dann, wenn die benötigten Ressourcen in der Umwelt vorhanden sind. Kann man eine solche Computer-Architektur realisieren? Wiesieht sie im Detail aus?

Versucht man es mit einer konventionellen Architektur, gerät man in Schwierigkeiten: Jede Erweiterung vergrößert räumlich das gesamte Rechensystem. Das verlangt zunehmend längere Drähte, insbesondere zum Taktgeber. Dies wiederum verlängert die Laufzeit der Signale; die Taktrate muss deshalb reduziert werden. Zudem vergrößert sich der fan-out (also die Anzahl Leitungen) des Taktgebers, dies vergrößert die Leistungsabgabe. Taktrate und Leistungsabgabe eines Schaltelementes können aber nicht beliebig gesteigert werden: Irgendwann bricht der Taktgeber zusammen.

Deshalb stellt sich die Frage nach einer prinzipiell unbeschränkt erweiterbaren Architektur *ohne* zunehmend längere Leitungen und *ohne* wachsenden fan-out. Petri beweist, dass dies in der Tat prinzipiell generell möglich ist, wenn jeder Erweiterungsschritt eine neue Komponente ausschließlich lokal an die vorherige Komponente anbaut. Diese Konstruktion hat aber ihren Preis: jede Komponente muss selbstständig aktiv sein können und das gesamte System arbeitet *asynchron*.

Die wesentliche Idee war der Entwurf eines asynchron arbeitenden Kellerspeichers, dessen Module jeweils einen Speicherplatz enthalten und sich nur mit ihren zwei Nachbarn bzw. einem Nachbarn und der Umwelt über die Speicherung oder Weiterleitung eines Objekts bzw. den Anbau eines weiteren Moduls verständigen müssen. Mit zwei solchen Kellerspeichern lässt sich ein in beide Richtungen beliebig verlängerbares Schieberegister erzeugen – und daraus eine universelle Turingmaschine mit den gewünschten Eigenschaften. Die Architektur eines solchen Schieberegisters hat Petri bereits im August 1962 auf dem ersten IFIP Welt-Computer-Kongress in München vorgestellt (Proceedings bei North Holland Publ. Comp. 1963).

Mit diesem Gedankenexperiment wollte Petri zeigen, dass asynchrone Systeme prinzipiell leistungsfähiger sind als synchrone. Daraus zog er den Schluss, dass jede allgemeine Theorie der Informationsverarbeitung – insbesondere wenn sie praxisrelevant sein soll – mit asynchronen, in ihrer Wirkung lokal begrenzten Operationen beginnen muss. Deshalb sei es auch ganz unangemessen, die Theorie der Informationsverarbeitung auf sequentiellen Modellen aufzubauen, mit verteilten Systemen als – gelegentlich nötiger – Verallgemeinerung.

Im Verlauf der Arbeit verwendet Petri eine Reihe formaler Notationen zur Beschreibung solcher asynchron arbeitenden verteilten Systeme: teils Graphen, teils Formel ausdrücke mit einem Parallelitätsoperator (wie sie viel später bei den Prozessalgebren verwendet wurden). Außerdem werden die für die Petri netze grundlegenden Begriffe der "Stelle" (zur Beschreibung lokaler Zustände) und der "Transition" (für lokal wirkende Aktionen, Ereignisse etc.) eingeführt.

Petri netze - so wie man sie jetzt kennt - erschienen zum ersten Male 1965 in Petris Vortrag "Grundsätzliches zur Beschreibung diskreter Prozesse" beim "3. Colloquium über Automatentheorie" in Hannover (publiziert 1967 im Tagungsband bei Birkhäuser). Bereits Ende 1964 sah der bekannte Software-Pionier Tom DeMarco Petri-Netze bei den Bell Telephone Laboratories im ESS-1-Projekt (in dem "the world's first commercial stored program telephone switch" entwickelt wurde); er war Mitarbeiter im Simulationsteam des Projekts und schreibt (im Springer-Buch "Software Pioneers, Contributions to Software Engineering"): "Among the documents describing the simulation was a giant diagram that Ms. Hoover called a Petri Net [Ms. Erna Hoover leitete das 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..." Etwas später schreibt DeMarco "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 Koder, observed that the diagram was a better spec than the spec".

Es ging Petri aber in seiner Dissertation um mehr. Durch die gesamte Arbeit zieht sich der Anspruch an die gesuchte Modellierungstechnik, nicht nur asynchrone verteilte Systeme angemessen zu modellieren, sondern zugleich eine Reihe weiterer Bedingungen zu erfüllen.

Zunächst einmal soll die Modellierungstechnik im Einklang mit der Physik stehen. Dazu gehört insbesondere, die Fiktion globaler Zustände aufzugeben. Jede diskrete Aktion eines endlichen Systems betrifft ja im Allgemeinen nicht alle seine Komponenten sondern nur einige wenige. Ein offensichtliches Beispiel ist ein Rechenschritt eines in das Internet eingebundenen PC's: Es wäre wenig sinngerecht, einen solchen Schritt als Änderung eines globalen Zustandes des gesamten Internets zu beschreiben. Petri schlägt vor, diesen Aspekt der Lokalität von Aktionen bei der Beschreibung von Systemabläufen genau zu modellieren und bei der Analyse und Verifikation eines System-Modells nur über die wirklich involvierten Komponenten zu argumentieren. Die Beschreibung einer Aktion als Paar aus altem und neuem (Gesamt-)Zustand gibt diese Information nur implizit preis. Erst recht nicht angemessen ist die Beschreibung eines einzelnen Systemablaufs als Sequenz von Aktionen. Statt Aktionen in der Zeit zu ordnen, schlägt Petri vor, sie nach ihren Ursache-Wirkungs-Zusammenhängen zu ordnen. Dabei können zwei Aktionen a und b durchaus ungeordnet bleiben, nämlich genau dann, wenn keine von beiden Resultate erzeugt, die die andere verwendet. Derart unabhängige Aktionen können "gleichzeitig" ("nebenläufig" in Petris Terminologie) ablaufen. Allerdings kann es sein, dass a unabhängig von b , und b unabhängig von c eintritt, obwohl a kausal vor c eintritt: Im Gegensatz zur "Gleichzeitigkeit" ist die Unabhängigkeit zwischen Aktionen nicht notwendig eine transitive Relation. Ähnlich ordnet die Relativitätstheorie jedem Ereignis e einen "Vor-Kegel" und einen "Nach-Kegel" von Ereignissen zu, die kausal vor bzw. nach e liegen. Indem die Kegel verschiedener Ereignisse sich überschneiden, entsteht eine Halbordnung. Die kausale Struktur von Ereignissen aus der Physik in die Informatik zu übertragen, ist ein besonders eindrückliches Beispiel für Petris Vorschlag, die Theorie der Informatik im Einklang mit der Physik zu bilden.

Zum Zweiten sollte die Modellierungstechnik in der Tradition naturwissenschaftlicher Theoriebildung stehen. Petri meint damit: Tiefliegende naturwissenschaftliche Theorie basiert auf Invarianten (beispielsweise Stoffgleichungen in der Chemie, Energieerhaltung in der Physik). Elementare diskrete Aktionen sollten einer entsprechenden Invarianz genügen. Notwendig dafür ist ihre *Reversibilität*: In einem Schritt $S \xrightarrow{a} S'$ einer Aktion a von einem (beschränkten, lokalen) Zustand S zu einem Zustand S' soll nicht nur S' aus S und a ableitbar sein, sondern umgekehrt auch S aus S' und a . Dies ist beispielsweise für die Wertzuweisung $x := x + 1$ der Fall, nicht aber für $x := 1$. Transitionen von Petrinetzen sind so gebildet, dass ihre lokale Ursache und Wirkung hervortritt und dass sie reversibel sind.

Als dritte Bedingung sollte die Modellierungstechnik nicht nur das Implementierbare beschreiben können, sondern auch den pragmatischen Umgang von Menschen mit Rechensystemen. Das begründet auch den Titel "Kommunikation mit Automaten", der ja durchaus doppeldeutig ist: Zum einen ist die Kommunikation zwischen Menschen und Automaten gemeint, zum anderen die Kommunikation zwischen Menschen *mit Hilfe von* Automaten.

Diese Ziele und die Ideen, um diese zu erreichen, konnten in der Dissertation nur skizziert werden. Diese Arbeit ist sicher keine Dissertation im gewöhnlichen Sinn und war es damals erst recht nicht – sie war eher zu sehen als ein Programm für die gerade entstehende Wissenschaft "Informatik". Ein Programm allerdings, das den damals gängigen Vorstellungen in seiner langfristigen Zielsetzung fremd war und in seinen kurzfristigen Vorschlägen den gängigen Vorstellungen zuwider lief – und auch nicht sehr leserfreundlich geschrieben war. So kann man sich lebhaft vorstellen, dass die Arbeit einiges Kopfzerbrechen verursacht hat. Der Darmstädter

Professor Alwin Walter, damals ein führender Pionier der ersten Generation des elektronischen Rechnens, hat die Bedeutung von Petris Arbeit erkannt und dafür gesorgt, dass sie als beste Dissertation des Studienjahres 1961/62 ausgezeichnet wurde.

In den USA wurde die Dissertation Mitte der 1960er Jahre zweimal ins Englische übersetzt: im Rahmen des Projekts MAC des MIT und in Anatol W. Holts "Information Systems Theory Project".

2. Carl Adam Petri

Verteiltheit mit Nebenläufigkeit als Grundlage der Informatik, Theoriebildung im Einklang mit der Physik, Invarianten in Analogie zu den Naturwissenschaften, formale Modellierung der Pragmatik des Umgangs mit Rechensystemen – das waren Themen, die der beginnenden Wissenschaft der Informatik gänzlich fremd waren und die kaum jemand in einer Dissertation jener Zeit erwartet hätte.

Was ist das für ein Mensch, der so unabhängig von aktuellen Moden und Strömungen wie selten jemand seine wissenschaftliche Vorstellungswelt aufbaut? Petris biographische Eckdaten sind typisch für seinen Jahrgang und schnell berichtet; zugleich aber erkennt man eine Kontinuität von für ihn typischen Erlebnissen seiner Kindheit und Jugend, hin zu seiner späteren wissenschaftlichen Arbeit.

Geboren 1926 in Leipzig, 1944 mit einem "Not-Abitur" der berühmten Thomas-Schule zum Militär gezwungen, geriet er in britische Gefangenschaft, blieb danach bis 1949 in England und studierte anschließend in Hannover Mathematik. Sein Lehrer Heinz Unger nahm ihn 1956 als Assistent mit nach Bonn. Nach seiner Promotion 1962 hat er das Rechenzentrum der Universität Bonn eingerichtet und geleitet, bei der Gründung der "Gesellschaft für Mathematik und Datenverarbeitung" (GMD) in Biringhoven bei Bonn (inzwischen übergeleitet in Institute der Fraunhofer-Gesellschaft) das Institut für Informationssystemforschung aufgebaut und bis 1991 geführt. Einen Ruf der Universität Dortmund auf einen Lehrstuhl für Informatik hat er 1973 abgelehnt. Nach seiner Pensionierung 1991 hat Petri seine Ideen weiter ausgebaut und publiziert.

Es gibt vielerlei Berichte und Anekdoten, die erklären helfen, warum Petri ganz bestimmte Fragestellungen und methodische Ansätze so zielstrebig betont hat. Drei davon seien hier berichtet:

Petris Vater war promovierter Mathematiker, der noch Minkowski und Hilbert gekannt und seinem Sohn vieles erklärt hat. Als Kind und Jugendlicher war Carl Adam Petri an naturwissenschaftlichen Fragestellungen interessiert. Zum 12. Geburtstag bekam er aus der Konkursmasse einer Buchhandlung zwei dicke Lehrbücher der Chemie, die er durcharbeitete. Sein Vater konnte ihm eine Ausnahmegenehmigung besorgen, als Jugendlicher unbeschränkt die Zentralbibliothek in Leipzig zu nutzen. Dort hat er sich an Arbeiten von Einstein und Heisenberg versucht.

Als Flakhelfer der Luftwaffe sah er Offiziere, die mit einfachen Mitteln, einschließlich Gehör und Augenmaß, die Höhe, Entfernung und Geschwindigkeit anfliegender Flugzeuge an die Kanoniere weitergaben. Die Kombination aus Messen und Abschätzen und vor allem die Frage nach der Verantwortung für (zwangsläufig) auftretende Fehler, hat ihn sehr beschäftigt.

In seinen Jahren in England hat Petri vermessungstechnische Probleme gelöst, beispielsweise die Anlage konzentrischer Ellipsen auf hügeligem Gelände.

Alle drei genannten Fragestellungen, die Methoden der Naturwissenschaften, die pragmatischen Aspekte des fehlerbehafteten Messens und die Argumentation über geometrische Objekte finden sich in seinem späteren Arbeiter wieder.

3. Die Jahre bis 1980

Petris Dissertation fand zunächst in Deutschland kaum Beachtung. Das ist nicht verwunderlich, wenn man sich noch einmal das Umfeld verdeutlicht, in dem sie entstanden ist. Die Rechentechnik der Zeit bestand aus Großrechnern für numerische Berechnungen mit Lochstreifen und Lochkarten zur Ein-/Ausgabe. Globaler Takt war selbstverständlich, an die Vernetzung von vielen Rechnern hat niemand gedacht. Software wurde vorwiegend zur Lösung numerischer Probleme geschrieben, im Wesentlichen in der seit 1956 verbreiteten Sprache FORTRAN. Mit der Entwicklung von ALGOL 60 hatte eine systematische Theorie der Programmierung sequentieller Prozesse begonnen; Petris Vorschlag, die Ein- und Ausgabe von ALGOL 60 als asynchrone Prozesse zu präzisieren, fand keinen Anklang. Fehlende E/A-Standards haben später die Verbreitung von ALGOL 60 behindert. Die praktische Informatik hat sequentielle Systeme favorisiert, nachdem Analogrechner und asynchrone Schaltwerke schwer beherrschbar und trotz Parallelarbeit zu langsam erschienen.

Die theoretische Informatik konzentrierte sich auf sequentielle Automaten- und Rechnermodelle sowie auf damit berechenbare Funktionen und formale Sprachen und auf den Compilerbau. Daneben gab es höchst spekulative Aussagen der "Kybernetik" und der "künstlichen Intelligenz", teilweise ohne realistische Vorstellungen für ihre Verwirklichung (wie beispielsweise für den "General Problem Solver"). Petri dagegen hat versucht zu beschreiben, wie seine Vorschläge mit der damals verfügbaren Technologie realisierbar sind; mehr noch: wie die Technologie pragmatisch, für vernünftige Zwecke nutzbar ist.

Zusammengefasst: Für die Praxis kamen Petris Vorschläge zu früh und die Theoretische Informatik hat andere Fragestellungen verfolgt. Von der zunächst geringen Resonanz hat Petri sich jedoch nicht beirren lassen. Mit einigen Mitarbeitern und vielen Besuchern hat Petri in den 1960er und 1970er Jahren seine Vorstellungen ausgebaut und publiziert.

Unter anderem hat er in den 1960er Jahren eine Reihe grundlegender Konzepte formuliert und in seinem Formalismus verwendet, die dann später von anderen wieder entdeckt worden sind. Dazu gehören das Alternieren von lokalen Zuständen und Übergängen (später u. a. in *message-sequence-charts*), seiteneffektfreie Aktionen (später in funktionalen Sprachen) und Multimengen von Tupeln als Zustände und darauf die Operationen "nehmen" und "geben" statt "lesen" und "schreiben" (später z. B. der "tupel space" von LINDA und der *Chemical Abstract Machine*).

In seiner Dankesrede zur Verleihung des Turing-Awards 1991 sagte Robin Milner: "Much of what I have been saying was already well understood in the sixties by Carl Adam Petri, who pioneered the scientific modelling of discrete concurrent systems. Petri's work has a secure place at the root of concurrency theory."

Ein erster Durchbruch im Bereich der Theorie kam Ende der 1960er Jahre mit der Beachtung, die Petris Arbeit im Projekt MAC des MIT gefunden hatte. Dieses überaus erfolgreiche Projekt stand (unter anderem) für "Multiple Access Computer" und entsprach damit durchaus auch Petris Vorstellungen. In den 1970er Jahren hat sich daraus in den USA (und anschließend auch in Deutschland und anderen europäischen Ländern) eine Schule entwickelt, die Petrinetze (oft mit beschrifteten Transitionen) meist als Mechanismus zur Konstruktion formaler Sprachen aufgefasst hat und umgekehrt Petrinetzklassen über ihre Sprachklassen charakterisiert hat. Obwohl damit eine ganze Reihe interessanter Eigenschaften von Petri-Netzen entdeckt und analysiert werden konnten, hat Petri diese Ansätze nicht sehr geschätzt. Sie berücksichtigen nicht den Unterschied zwischen unabhängig und kausal geordnet eintretenden Ereignissen, auf den Petri so viel Wert gelegt hat. Es wurden aber auch eine Reihe Petrinetz-spezifischer Ideen, wie insbesondere die Verwendung der linearen Algebra zur Berechnung von Systeminvarianten entwickelt bzw. weiter ausgebaut, weil sie sehr wichtig für die Anwendungen sind.

Was es an theoretischen Arbeiten und praktischen Anwendungen gab, wurde 1979 in einem "Advanced Course" an der Universität Hamburg zusammengestellt und publiziert (LNCS Band

84). Der Kurs fiel in eine Zeit, in der verteilte Systeme insgesamt mehr Aufmerksamkeit fanden und in der auch mit anderen Ausdrucksmitteln an Theorien für solche Systeme gearbeitet wurde.

4. Die Zeit seit 1980

Seit Beginn der 1980er Jahre wuchs die Zahl der theoretischen und praktischen Arbeiten zu Petrinetzen sprunghaft an. Als ganz neues Konzept wurde die Markierung von Plätzen mit mehreren, gleichartigen "schwarzen" Marken auf verschiedene, individuelle, "bunte" Marken verallgemeinert – ein Schritt, der die Ausdruckskraft der Modellierungstechnik massiv erhöht und dabei die entscheidenden Analysetechniken, insbesondere die linear-algebraischen Kalküle der Platz-Invarianten und der Transitions-Invarianten zweckmäßig verallgemeinert.

In den 1980er Jahren trieb das wachsende Interesse an verteilten und reaktiven Systemen nicht nur die Forschung an Petrinetzen voran, sondern auch an einer Vielzahl von Alternativen. Vieles davon ist schnell wieder verschwunden. Zu den langfristig erfolgreichen Vorschlägen gehörten die bereits erwähnten Prozessalgebren, beispielsweise in den 1990er Jahren der π -Kalkül von R. Milner, aber auch Harels "Statecharts" und viele Varianten Temporaler Logik. Nebenläufigkeit (concurrency) und Kooperation prägen seither die Forschung an verteilten und reaktiven Systemen.

Wie weit beispielsweise die von Beginn an grafische Darstellung von Petrinetzen die grafischen Konzepte von SADT oder Statecharts beeinflusst hat, wird niemand sagen können. Klarer ist der Einfluss des Konzepts der oben beschriebenen verteilten Abläufe auf effiziente Techniken des Model Checking nachweisbar: Eine halbgeordnete Menge von Aktionen ist der Analyse mit temporaler Logik nicht unmittelbar zugänglich. Temporale Logik verwendet total geordnete Sequenzen von Schritten. Nun entspricht ein verteilter Ablauf A einer Menge solcher Sequenzen, nämlich allen Ergänzungen der Halbordnung von A zu einer totalen Ordnung. Zur Analyse einer Reihe von Eigenschaften von A reicht es nun, eine einzige, frei wählbare Sequenz zu untersuchen. Dies reduziert entscheidend die Komplexität des Model Checking.

Petrinetze tragen vorzüglich zum viel diskutierten, aktuellen Thema des modellbasierten Entwurfs bei. Viele Methoden haben Komponenten, die Konzepte von Petrinetzen nachbilden. Dazu gehören insbesondere auch die Aktivitäts-Diagramme der UML 2.

Als Technik zur Modellierung rechnerintegrierter Systeme sind Petrinetze nun seit langem etabliert. Eine große Wissenschaftlergemeinschaft und die Software-Industrie verwenden Petrinetze in ganz unterschiedlichen Projekten. Wenig bekannt in der Informatik ist die sehr hohe Wertschätzung der Petrinetze im Bereich der Technik, bei Ingenieuren. Bei der Verleihung des Werner-von-Siemens-Rings an C.A. Petri (als 30. "hochverdienter Naturforscher und Gestalter der Technik" und zweiter ausgezeichneten Informatiker nach Konrad Zuse) wurde hervorgehoben: "Bei den Ingenieuren führten Petrinetze zu einem Durchbruch in der Behandlung diskret gesteuerter Systeme. Petrinetze kommt bei der Lösung des Entwurfsproblems eine Schlüsselrolle zu, weil diese erstmals eine eindeutige Beschreibung und auch eine aussagekräftige Analyse diskret gesteuerter Systeme zulassen. Auf der Basis von Petrinetzen können nun auch für diskrete Steuerungssysteme System-Invarianten angegeben werden" (Prof. Gottzein).

Eine Vielzahl nützlicher, über einen "Petri Net Kernel" miteinander vernetzte Werkzeuge ist verfügbar, periodisch werden Konferenzreihen organisiert, insbesondere seit 1980 die jährliche "International Conference on Applications and Theory of Petri Nets" mit Tutorien, speziellen Workshops etc. Seit 1987 gibt es die GI-Fachgruppe "Petrinetze und verwandter Systemmodelle" mit fast 60 Ausgaben ihres umfangreichen Newsletters. Unter www.informatik.uni-hamburg.de/TGI/PetriNets/index.html verwaltet das Institut für Informatik der Universität Hamburg das "Petri Nets World"-Portal mit vielen aktuellen Verweisen auf Bibliographien, Werkzeuge, Veranstaltungen etc.

5. Ehrungen

Nachdem sich Petrinetze in den 1980er Jahren endgültig als Modellierungstechnik durchgesetzt hatten, ist Carl Adam Petri vielfach geehrt worden. Zu den herausragenden Auszeichnungen gehören:

- 1988 Verdienstkreuz 1. Klasse des Verdienstordens der Bundesrepublik Deutschland,
- 1988 Honorarprofessor der Universität Hamburg,
- 1989 Mitglied der Academia Europaea, London,
- 1993 Konrad-Zuse-Medaille der Gesellschaft für Informatik für besondere Verdienste um die Informatik,
- 1997 Werner-von-Siemens-Ring für herausragende Verdienste um die Technik in Verbindung mit der Wissenschaft,
- 1997 Mitglied der New York Academy of Sciences,
- 1999 Ehrendoktor der Universität Zaragossa,
- 2003 Commandeur in de Orde van de Nederlandse Leeuw.

6. Was bringt die Zukunft?

In seinem eingeladenen Vortrag auf der 26. International Conference on Application and Theory of Petri Nets in Miami im Juni 2005 hat Petri sich sehr erfreut geäußert über die Vielfalt und die Qualität von Anwendungen seiner Theorie. Er hat dennoch angemahnt, die Theorie prinzipiell und substantiell auszubauen. Nicht, indem man noch speziellere Petrinetz-Klassen und passende Analysealgorithmen entwickelt und charakterisiert, sondern indem man die großen, langfristigen Ziele aufgreift, die er schon in seiner Dissertation skizziert hat. Viele davon werden immer noch nicht bearbeitet. Vielleicht muss auch hier erst der Fortschritt der Hardware und der Druck der Anwendungen stark genug sein, um sich auf seine Vorschläge zu besinnen.

Ein Beispiel für solch ein Ziel ist die Suche nach *Erhaltungssätzen* der Informationsverarbeitung in Analogie zu Erhaltungssätzen der Physik (wie Einsteins $e = mc^2$) und der Chemie (etwa die Stoffgleichung $\text{NaJ} + \text{Cl} \rightarrow \text{NaCl} + \text{J}$). Vielleicht - so meint Petri - wird man zukünftig Software-Schnittstellen mit Hilfe solcher Erhaltungssätze sehr viel abstrakter und zugleich präziser formulieren können, als dies heute möglich ist.

Dazu fehlt ein grundlegender Begriff dessen, was bei dynamischen Vorgängen der Informatik erhalten bleibt. Ein Vorschlag dafür - meint Petri - wäre ein neuer Begriff von "Information". Wenn es die Aufgabe von Informatik-Systemen ist, Information zu *wandeln*, müsste dabei die Gesamtheit der Information erhalten bleiben. Kein derzeit gängiger Begriff von "Information" hat diesen Charakter. Wie der gesuchte Begriff tatsächlich aussehen könnte, ob es eine kleinste Informationseinheit gibt, ist völlig offen. Petri ist weiter auf der Suche danach.

Die Informatik hat sich stürmisch und ungeplant entwickelt, getrieben von ihrem technischen und ökonomischen Potential. Dies hat Auswirkungen auf die Wissenschaft, die Erfolg fast nur noch an Projekten misst, die auf Monate oder ganz wenige Jahre hin angelegt sind. In dieser Welt hätte Petri kaum eine Chance gehabt. Nur Wenige fragen wie er über den Tag hinaus, wie denn eine systematische Wissenschaft der Informatik aussehen könnte.

Wilfried Brauer und Wolfgang Reisig