

2 Petrinetze

Bisher wurden Petri-Netze als autarke Systeme betrachtet, die nicht in Kommunikation mit ihrer Umwelt stehen. Ihr dynamisches Netzverhalten wird lediglich von der Netzstruktur und der aktuellen Netzmarkierung bestimmt. Die Aufnahme beliebiger Petri-Netz-Elemente lässt zwar die Modellierung der Beziehungen zur Umwelt zu, doch tatsächliche Kommunikation findet nicht statt. Für die Steuerungstechnik ist jedoch gerade Kommunikation zur Umwelt über die Steuerungssperipherie integraler Bestandteil. Ein Kommunikationsmechanismus für die Nutzung der Petri-Netze zur Prozesssteuerung ist unerlässlich. Solche Petri-Netze bedürfen einer steuerungstechnischen Interpretation. Interpretierte Petri-Netze besitzen eine Struktur Erweiterung zu herkömmlichen Petri-Netzen. Die einfachste Form der Erweiterung, die zu interpretierten Petri-Netzen führt, ist eine Ankopplung externer Ereignisse sowie die Ausgabe von Signalen an das ansonsten autonome Petri-Netz – gelegentlich auch synchronisierte Petri-Netze genannt. Die hier vorgestellten Erweiterungen gehen über solche Ankopplung hinaus. So entstanden einige Kantenarten, die Funktionen zur Verfügung stellen, die typisch für die Steuerungstechnik sind (Flankenerkennung und Negationen). Oft ist es hilfreich, wenn eine Integration von Boole'scher Algebra und textueller Source in ein Petrinetz stattfinden kann. So lassen sich die Vorteile aller drei Beschreibungsformen innerhalb eines Modells nutzen, und Einschränkungen für die Modellentwicklung fallen minimal aus.

2.1 Petrinetz-Eigenschaften – Überblick und Beispiel

Die Einfachheit und daraus resultierende Verständlichkeit, die Petrinetze auszeichnet, stammt nicht nur von den wenigen Elementen, die ein Petrinetz bilden, sondern auch von den in allen Netzklassen wiederkehrenden einfachen Prinzipien, die ein Netzelement repräsentiert, und der hohen Transparenz, die die Bild realer Prozesse mittels Petrinetzen aufweist. Die folgende kurze Beschreibung soll dies nochmals verdeutlichen.

Von einem Prozesszustand in den nächsten gelangt man nur, wenn die zugehörige Bedingung erfüllt ist. Die Petrinetze spiegeln genau dieses Verhalten wider, so dass im Grunde alle Prozesszustände mit den Übergängen und deren Bedingungen in der richtigen Reihenfolge bereits das Netz darstellen.

Die Plätze repräsentieren die Prozesszustände. Die Übergänge werden durch die Transitionen dargestellt. Die einlaufenden (Pre-)Kanten und deren Verknüpfungen (UND oder ODER) an den Transitionen beinhalten die Übergangsbedingungen (Schaltbedingungen).

Demzufolge besteht ein Petrinetz aus einer Aneinanderreihung von Plätzen und Transitionen, die durch die gerichteten Kanten verbunden sind. Analog besteht der abgebildete Prozess aus einer Aneinanderreihung von Prozesszuständen (Situationen) und Zustandsübergängen (Ereignisse), die durch die Prozesseigenschaften in direkter Beziehung stehen.

2.2 Sprachbetrachtung

Eine der wesentlichen Forderungen der Automatisierungstechnik hinsichtlich Soft- und Hardware ist die garantierte Antwortzeit bei geringem Ressourcenbedarf (HW-Kosten). Hieraus besteht die Forderung nach hoher Ausführungsgeschwindigkeit bei geringem Speicherbedarf (generierte temporäre und versteckte Daten) und Kompaktheit des Codes. Dies gilt insbesondere für grafische Sprachen, aus denen letztendlich ein textuelles Äquivalent mit diversen, vor dem Anwender versteckten zusätzlichen Variablen generiert wird.

Ist ein Petrinetz in einer SPS-Sprache umzusetzen, sind die Eigenschaften dieser Sprache zu benutzen. Grundsätzlich gibt es nur wenige Anforderungen, die die Petrinetze an eine Sprache stellen. Sprachen wie AWL (IEC 61131, S5, S7 usw.), Assembler, ST, Java, Pascal, C, Neuron C usw., um nur einige zu nennen, sind geeignet. Eine Analyse der Spracheigenschaften ergibt, dass einige Sprachen hinsichtlich der Erzeugung eines Codes mit kurzen Reaktionszeiten besser geeignet sind als andere. Steht ein Petrinetz-Editor zur Verfügung, d. h. Codetransparenz steht nicht im Vordergrund, der die Code-Erzeugung übernimmt, sind Spracheigenschaften wie Sprünge und Unterprogrammaufrufe hilfreich. Aus diesen und weiteren Gründen sind alle im Folgenden gezeigten Beispiele einer Codierung bevorzugt in AWL und IEC-61131-2nd-Edition-konform. In Ausnahmen wird die Codierung in ST nach IEC-61131-2nd-Edition gezeigt.

Dabei ist stets von einer Realisierung in den Programmorganisationseinheiten (POE) Funktionsbaustein oder Programm auszugehen, sofern dies nicht bereits aus den Beispielen hervorgeht. Wegen der temporären Datenhaltung eignen sich Funktionen nur in wenigen Spezialfällen für die Aufnahme von Petrinetzen. Deshalb bleibt dieser Aspekt im Weiteren unberücksichtigt.

2.3 Ablaufsprachen (SFC und Petrinetze)

2.3.1 Ablauforientierte Programmierung, IEC-61131-konform

Dieser Beitrag soll kein Votum gegen die IEC 61131 sein, im Gegenteil, vielmehr soll hier auf Methoden und Konzepte der Steuerungstechnik aufmerksam gemacht werden, die dem Nutzen und dem Vorteil des Steuerungstechniklers in Bezug auf Konzeption, Inbetriebnahme, Wartung, Kompatibilität und Portierbarkeit dienen. Dazu gehört eine Analyse des Ist-Zustands sowie die Beurteilung der Spracheigenschaften hinsichtlich Eignung und Transparenz.

Methodische Vorgehensweisen und Verfahren sind in den Ingenieursdisziplinen Basis des Erfolgs. Methoden der Steuerungstechnik lassen sich in zwei unterschiedliche Klassen einteilen. Die Verknüpfungssteuerung basiert im weitesten Sinne auf der Boole'schen Algebra und die Ablaufsteuerung auf der Folge bzw. dem Ablauf von aufeinander folgenden Schritten bzw. Zuständen. Ideale Unterstützung für die Realisierung steuerungstechnischer Aufgaben eines Projekts ist häufig ein Mix beider Methoden sowie die Möglichkeit, die SPS-Sprachen gemäß ihren Eigenschaften aufgabenbezogen und zielgerichtet einsetzen zu können. Damit haben beide ihre Berechtigung. Letztere werden in der Praxis meist seltener eingesetzt. Ob es an unzureichenden Tools, wenig praxisgerechter Auslegung der Theorie, Ausbildungsmangel, Unkenntnis, Firmenstandards oder anderem liegt, muss jeder für sich herausfinden.

Neben der Steuerungstechnik sind vielfach auch regelungstechnische Aufgaben zu bewältigen, diese lassen sich ebenfalls in zwei Klassen gliedern. Die klassische Regelungstechnik geht von einer mathematisch beschreibbaren Regelstrecke aus, während die Fuzzy-Technologie auf einem Regelwerk basiert, das aus Erfahrungen heraus gebildet wird. Beide, Ablaufsteuerung und Fuzzy-Control, spielen immer noch die kleinere Rolle. Jedoch steigen ihre Anwendungsfälle deutlich an. Es wird erkannt, dass nicht nur reine Philosophiefragen hinter der Technik stehen, sondern dass sich diese in Euro und Cent rechnet, insbesondere bei ganzheitlicher Betrachtung des gesamten Lebenszyklusses.

Viele Abhandlungen bleiben bei mathematischen, analytischen und theoretischen Betrachtungen. Tools basieren oft auf diesen und kommen häufig nicht über das Stadium von Diplomarbeiten hinaus, die nicht zuletzt, weil die Synthese aus Theorie und Praxis, d. h. die Anwendbarkeit, nie vollzogen wurde, oder anfänglich in der Entwicklung unberücksichtigt blieb. Ein nachträgliches Aufpflöpfen führt häufig zu Qualitätsproblemen und mangelnden ergonomischen Eigenschaften.