

## 1 Motivation

Zwischen der klassischen Informatik und der Automatisierungstechnik bestehen enge Beziehungen, wie auch einige Besonderheiten. Die Gemeinsamkeiten resultieren vor allem aus der Benutzung des immateriellen Mediums Information und der Verwendung informationsverarbeitender Einrichtungen in Gestalt von Computern, wenn auch in verschiedenartigen Ausführungen. Funktionsbestimmend sind in beiden Fällen Programme, welche jedoch Unterschiede in der Programmstruktur aufweisen. Die größten Besonderheiten resultieren jedoch aus den unterschiedlichen Betriebsarten.

Die nachfolgenden Ausführungen sind dazu bestimmt, die Aspekte von Steuerung (*control*), Kommunikation (*communication*) und Informationsverarbeitung (*computing*) im Sinne der Wiener'schen Definition der Disziplin „Kybernetik“ \*) unter dem Gesichtspunkt der Informationsprozesse zusammen zu führen.

## 2 Gegenstand der Prozessinformatik

Ausgangspunkt unserer Betrachtung ist der *Prozess*. Unter einem Prozess versteht man allgemein das Vorhandensein von Vorgängen in einem abgeschlossenen System, die zu Veränderungen in Raum und/oder Zeit führen. Für unsere Zwecke treffen wir noch eine Unterteilung in Rechenprozesse und technologische Prozesse. *Rechenprozesse (computational processes)* beinhalten eine Folge von sequentiell abzuarbeitenden Aktivitäten, durch die eine abgeschlossene Aufgabe bearbeitet wird. Unter *technologischen Prozessen (technological processes)* versteht man Folgen von Aktivitäten in technischen Anlagen oder einem technischen Gerät zur Erfüllung einer bestimmten Aufgabe. Damit technologische Prozesse einen Nutzen erbringen, müssen diese zielorientiert geführt werden. Somit wird eine Steuerung benötigt, deren Funktion in unserem Fall von einer rechnerbasierten Lösung erbracht wird. Damit stoßen wir wieder auf die bereits betrachtete Klasse der Automatisierungssysteme.

Die technologischen wie Rechenprozesse können häufig hinsichtlich der zu bearbeitenden Teilaufgaben untergliedert werden, was dann zur Bildung von *Teilprozessen* und somit zu einer Modularisierung führt. Hier werden die Bausteine von Rechenprozessen als *Tasks* bezeichnet. Die Realisierung komplexer Aufgabenstellungen macht eine wohl organisierte Abstimmung der Tasks erforderlich, wofür eine Taskverwaltung benötigt wird. Die Abarbeitung der einzelnen Tasks erfolgt dabei schrittweise (sequentiell). Es gibt auch Tasks, die parallel wirken und somit (Prozess-) *Nebenläufigkeiten* erzeugen. Diese können ihrerseits wiederum mehrere sequentielle Teilprozesse enthalten. Nebenläufige Tasks müssen wieder zusammengeführt und damit in eine Reihenfolge gebracht werden, was eine *Koordination* besorgt. Für das Zusammenspiel der verschiedenen Tasks sind die Prinzipien von Kooperation oder Konkurrenz maßgebend. Bei *Kooperation* sind die Tasks miteinander verkettet, wobei ein Teilprozess erst dann starten kann, wenn der vorgelagerte Prozess seine Ergebnisdaten geliefert hat. Die Situation einer *Konkurrenz* besteht hingegen, wenn die Aktivitäten eines Prozesses die eines anderen zu behindern drohen. Ein konkurrierender Teilprozess kann daher erst gestartet werden, wenn

ein bereits laufender Prozess beendet ist, um Kollisionen zu vermeiden. Es ist daher notwendig, sowohl kooperierende als auch konkurrierende Teilprozesse miteinander zu synchronisieren.

Für die Arbeit der Tasks ist es notwendig, dass diese über Prozessgrenzen hinweg mit Daten bzw. Datenblöcken versorgt werden. Dies kann durch den Zugriff auf gemeinsame Datenbestände oder durch Datenübergabe zwischen aufeinander folgend zu bearbeitenden Teilprozessen, also Intertaskkommunikation, erfolgen. In bestimmten Fällen muss der Datenzugriff jedoch beschränkt werden, etwa wenn es sich um sicherheitsrelevante oder personenbezogene Daten handelt. In solchen

---

\*) N. Wiener: Cybernetics or Control and Communication in the Animal and the Machine

Fällen sind programmtechnisch besondere Schutzmaßnahmen vorzusehen. Bei weniger sicherheitsgefährdeten Anwendungen genügt es zumeist, die zu übertragenden Daten mit einem Nutzungshinweis zu verbinden. Auf diese Weise kann auch eine implizierte Synchronisation erreicht werden. Kommunikationsprozesse spielen auch beim Dialog von Prozessrechnern mit technologischen Prozessen eine besondere Rolle, worauf im folgenden Abschnitt näher eingegangen wird.

Die zeitliche Abstimmung hinsichtlich der Bereitstellung und Nutzung von Daten oder auch Ressourcen wird durch *Synchronisation* erreicht. Zur Verhinderung von Zugriffskonflikten werden sog. *Semaphore* eingesetzt. Dieses aus dem sportlichen Staffellauf bekannte Verfahren gewährleistet, dass die Aktionen sich beeinflussender Tasks in eine zeitlich geordnete Abfolge gebracht werden. Mittels Semaphore lassen sich auch Warteschlangen bilden, in die sich mehrere Tasks durch Anmeldung einreihen können. Nach Bedienung der aktuell bearbeiteten Task wird dann die Warteschlange um jeweils einen Platz verkürzt. Zur Erfassung von Bedingungen können auch Ereignisflags verwendet werden, die wie binäre Semaphore wirken.

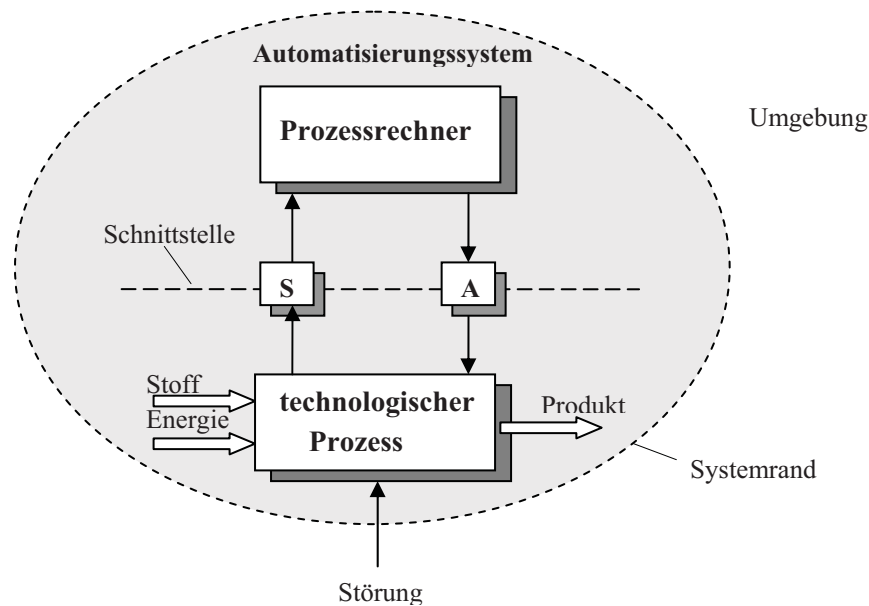
### 3 Besonderheiten des Einsatzes von Prozessrechnern

Prozessrechner finden bei der Steuerung vorzugsweise von technologischen Prozessen Verwendung. Solche Prozesse realisieren sich in technischen Anlagen, in denen ein Produkt mit bestimmten Eigenschaften (bspw. Dieselkraftstoff oder ein Fahrzeugteil) hergestellt wird, oder in technischen Geräten (bspw. einem Drucker). Die Aufgabe eines Prozessrechners besteht dann in der selbsttätigen Führung solcher Prozesse im Hinblick auf ein vorgegebenes Ziel. Prozessrechner sind somit das bevorzugte Mittel der Automatisierungstechnik.

Aus dem Einsatz von Computern zur Prozesssteuerung resultieren einige Besonderheiten. Zur Verdeutlichung stellen wir die Betriebsart von Prozessrechnern derjenigen klassischer Computer gegenüber. Bei der üblichen Nutzung von Computern – etwa im Verwaltungs- und Bürobereich, zur Datenauswertung, beim Entwurf, der Durchführung wissenschaftlicher Berechnungen und vielen anderen Zwecken – findet ein mehr oder weniger sporadischer Dialog zwischen dem Computer und menschlichen Nutzer statt. Bei Gebrauch des Computers nimmt der Benutzer zunächst einige Eingaben (Parameter, Anfragen etc.) vor, wonach der Computer die Aufgabe programmgemäß löst und das Ergebnis dem Benutzer in geeigneter Form präsentiert. Zwischen Programmstart und Ergebnisausgabe besteht ein Zeitabstand, der je nach Arbeitsgeschwindigkeit und Programmkomplexität in weiten Grenzen variieren kann.

Im Gegensatz dazu steht der Prozessrechner in einem permanenten Dialog mit einem vorzugsweise technologischen Prozess. Der Mensch übt hier zumeist nur eine überwachende Funktion aus oder übermittelt dem Rechner ab und an gewisse Vorgaben. Beim beständigen Zusammenspiel von Prozessrechner und technologischem Prozess werden wechselseitig Informationen ausgetauscht. Auf diese Weise besteht ein sich ein geschlossener Informationskreislauf. Diese Betriebsart ist typisch für Steuerungs- und Regelungsprozesse und wird *on-line-Betrieb* genannt. Bei anderen Prozessrechneranwendungen – etwa zur Prozessüberwachung – fehlt der Informationsrückfluss vom Rechner an den Prozess. In diesem Fall liegt ein *off-line-Betrieb* des Prozessrechners vor. Adressat der Computerergebnisse ist hier der Mensch. Der Operateur wird durch die Tätigkeit des Prozessrechners über den Prozessverlauf informiert und auf kritische Situationen aufmerksam gemacht. Bei Eintreten eines Störfalls muss er selbst Entscheidungen über entgegenwirkende Maßnahmen treffen und diese auch realisieren. Systeme beider Strukturen sind in der Automatisierungstechnik anzutreffen.

Die Struktur des umfassenderen Automatisierungssystems mit Steuerungsfunktion veranschaulicht das folgende Bild.



Struktur eines Automatisierungssystems mit Prozessrechner

Das Automatisierungssystem ist gegenüber der Umgebung durch einen Systemrand abgegrenzt. Der Prozessrechner empfängt fortlaufend aktuelle Informationen über den Prozesszustand in Form von (Mess-)Daten und verarbeitet diese prozess-schritthaltend zusammen mit gespeicherten Daten entsprechend einem aufgabenspezifischen Anwendungsprogramm zu neuen Daten. Damit sich der Prozessrechner (informationelles Teilsystem) und der technologische Prozess (physisches Teilsystem) verständigen können, bedarf es einer speziellen Peripherieausstattung. Die benötigten Schnittstellenelemente sind auf der einen Seite die Sensoren **S**, welche Werte relevanter Messgrößen liefern. Auf der Gegenseite werden die berechneten Stellgrößenwerte mit Hilfe von Aktoren **A** in Handlungen zur Beeinflussung von Stoff- oder Energieströmen umgesetzt, um damit auf den technologischen Prozess zielgerichtet einzuwirken.

Prozessrechner kommunizieren mit dem technologischen Prozess typischerweise durch den Austausch von *Signalen*. Je nach Wirkprinzip der Sensoren bzw. Aktoren können die Signale unterschiedlichen Charakter haben, wobei vor allem analoge, binäre und codierte Signale auftreten. Da der Prozessrechner nur Daten verarbeiten bzw. liefern kann, werden entsprechende Umsetzer benötigt. Des Weiteren wird ein periodisch arbeitender Unterbrechungsmechanismus (Signalgeber) benötigt, um aus kontinuierlich übertragenen Signalen Daten herzustellen. Für die Signalübertragung werden die klassischen Übertragungskanäle benutzt. Zur Abwehr von Signalstörungen werden vor allem verdrehte Kupferleitungen, Koaxialkabel und auch Lichtwellenleiter eingesetzt. Bei verteilten Automatisierungssystemen finden verbreitet auch multiplex betriebene Übertragungsmedien in Form von (Prozess-)Bussystemen Anwendung.

Eine wesentliche Besonderheit des Betriebs von Prozessrechnern besteht darin, dass die Verarbeitungsergebnisse innerhalb einer durch die Dynamik der jeweiligen technologischen Prozesse vorgegebenen Zeit bereit gestellt werden müssen. Dies verlangt einen *Echtzeitbetrieb (real-time processing)*, wobei u. U. harte Zeitanforderungen zu erfüllen sind. Zeitbedingungen spielen bei Prozessrechnern auch in anderer Form eine wichtige Rolle. Dazu gehört die Bildung von *Relativzeiten* in Form von Zeitverzögerungen, Wartezeiten oder Zeitfenstern. Auch *Absolutzeiten* werden benötigt, um beispielsweise der Aufzeichnung von Prozessdaten oder dem sporadischen Auftreten störrelevanter Prozessereignisse einen zeitlichen Bezug zuzuordnen oder wenn zu festgelegten Zeiten bestimmte Handlungen ausgelöst werden sollen. Besonders bei den Kommunikationsprozessen ist wiederum eine Zeitüberwachung vorzusehen, um zu verhindern, dass Rechenprozesse beim Nichteintreten eines erwarteten Ereignisses blockiert werden. Tritt dieses Ereignis nicht innerhalb der Karenzzeit ein, so wird eine Fehlerbehandlung veranlasst. Zur Realisierung all dieser zeitabhängigen

Anforderungen benötigen Prozessrechner einen Zeitgeber in Form einer *Echtzeituhr*, aus deren Signal die benötigten Zeitoperationen abgeleitet werden.

Ein weiteres Merkmal des Prozessrechnereinsatzes ist die Verarbeitung spontan auftretender Prozessereignisse – etwa kritischer Prozesszustände. Dazu werden vom Rechner sehr kurze Antwort- bzw. Reaktionszeiten verlangt. Beim Melden solcher Ereignisse ist eine Unterbrechung des laufenden Programms durch einen *Interrupt* erforderlich. Dieser veranlasst, dass die Warnmeldung bevorzugt durch eine spezielle Routine bearbeitet wird. Oftmals sind Prozessereignisse mit gestufter Dringlichkeit zu berücksichtigen, so dass ein mehrstufiges und hierarchisch strukturiertes *Prioritätskonzept* benötigt wird.

Für die in Prozessrechnern ablaufenden Rechenprozesse gelten weitgehend die bereits in Abschn. 2 aufgeführten Merkmale. Dazu zählt die Aufgliederung in kooperierende wie auch konkurrierende Teilprozesse. Die auftretenden Rechenprozesse haben überwiegend sequentiellen Charakter, sind aber vielfach auch nebenläufig. Die wechselseitigen Abhängigkeiten der Systemtasks sind hier allerdings deutlicher ausgeprägt. Außerdem bestehen erhöhte Anforderungen an den Interruptbetrieb angesichts der vielfach in großer Zahl zu berücksichtigenden äußeren Ereignisse. Die Software-Struktur von Prozessrechnern korrespondiert weitgehend mit dem modularen Aufbau der Hardware und ist dementsprechend in eine Vielzahl von Teilprogrammen gegliedert. Die Verwaltung der Tasks, Interrupts, Inputs und Outputs übernimmt ein *Echtzeit-Multitasking-Betriebssystem (real-time multitasking operational system)*.

Auch die vorstehenden Ausführungen zur Implementierung der Rechenprozesse haben hier Gültigkeit. Neben der prinzipiell möglichen konzentrierten Hardware finden vor allem modulare und damit funktionell verteilte Lösungen Anwendung. Dementsprechend sind die früher vorherrschenden großen kompakten Prozessrechner weitgehend durch eine dezentrale und weitgehend autonom agierende Hardware ersetzt, welche in ein Netz eingebunden sind. Die Gründe dafür liegen neben der besseren Transparenz und Erweiterbarkeit vor allem in der höheren Verfügbarkeit solcher Automatisierungslösungen. So finden sich – um ein Beispiel zu geben – in einem PKW der gehobenen Mittelklasse je nach Ausstattung bis zu 60 verteilte Prozessoren, welche zur Lösung der verschiedensten Betriebs- und Sicherheitsaufgaben eingesetzt sind und über ein Bussystem kommunizieren.

Ein weiteres Spezifikum des Prozessrechnereinsatzes resultiert aus dem Dauerbetrieb in technologischen Prozessen. Dementsprechend stellen sich hohe Anforderungen an die *Verfügbarkeit* der Einrichtung. In besonders sicherheitssensiblen Einsatzfällen – etwa in Atomkraftwerken – müssen besondere Maßnahmen zur Gewährleistung eines hochgradigen Sicherheitsstandards getroffen werden. Diese bestehen in der Einführung von Redundanzen, beispielsweise in Form der Mehrfachausrüstung von Komponenten, Back-up-Lösungen sowie dem Einsatz von Parallelrechnern mit gegenseitiger Überwachung.

Aus der Verwendungsart von Prozessrechnern resultieren auch besondere *Einsatzanforderungen*. Davon betroffen sind vor allem die Peripheriekomponenten, manchmal aber auch die Prozessoren, da diese in Prozessnähe angeordnet und somit u. U. erheblichen Umgebungseinflüssen ausgesetzt sind. Extreme Einsatzanforderungen bestehen beispielsweise bei der Verwendung von Prozessrechnerlösungen im Weltall.

Die bereits genannten Prozessmerkmale haben auch bei den zu steuernden technologischen Prozessen Gültigkeit. Technologische Prozesse sind zumeist in eine Anzahl von Teilprozessen aufgliedert, die miteinander in Wirkbeziehungen stehen. Das Zusammenspiel kann dabei in kooperativer oder auch konkurrierender Weise erfolgen. Kooperatives Verhalten ist beispielsweise zu beobachten, wenn in einer Anlage Produkte mit vorgegebenen Merkmalen in aufeinander folgenden Bearbeitungsschritten hergestellt werden. Ein Beispiel für Konkurrenz bietet hingegen der Verkehr mehrerer sich in einer Fabrikhalle autonom bewegender Transportfahrzeuge (*carrier*), die sich eine gemeinsame Ressource

(hier: den Fahrweg) teilen. Die Steuerungen solcher funktionell verteilter Prozesse müssen folglich koordiniert werden.

Technologische Prozesse können auch in räumlich verteilten Anlagen stattfinden. Die Führung solcher dezentralisierter Prozesse verlangt dann auch den Einsatz lokaler Steuerungen. Somit entstehen funktionale Einheiten, die nebenläufig sind und völlig autonom agieren. Die Zusammenführung der Prozesssteuerungen dezentralisierter technologischer Prozesse zu einem gesteuerten Gesamtprozess mit zentraler Zielstellung führt dann zu hierarchischen Automatisierungsstrukturen. Innerhalb dieser werden Informationen, ggf. über mehrere Ebenen hinweg, in Form von Signalen ausgetauscht. Ein typisches Beispiel sind zur Erzeugung von Elektroenergie aus regenerativen Quellen eingesetzte verteilte Windkraftwerke.

Computer finden in der Automatisierungstechnik auch als Hilfswerkzeuge Verwendung. Hierbei handelt es sich um hochleistungsfähige *Tools* (beispielsweise Werkzeugkiste von *MATLAB*, *FuzzyTech* u. a.), die auf klassischen Rechnern laufen. Diese Werkzeuge sind nicht nur für Standardaufgaben einsetzbar, sondern erleichtern den Automatisierungsingenieuren auch die Handhabung anspruchsvoller und komplexer Automatisierungsverfahren. Sie unterstützen außerdem die Systemidentifikation (Bildung von Prozessmodellen) und ermöglichen eine Systemsimulation vor Inbetriebnahme des Automatisierungssystems.

Bei der Umsetzung des Systementwurfs in ein Anwenderprogramm werden bevorzugt modular strukturierte *Fachsprachen* eingesetzt. Damit kommt man dem ingenieurtechnischen Personal entgegen, welches damit die Programmerstellung in der gewohnten Denk- und Arbeitsweise durchführen kann. Besondere Bedeutung haben Fachsprachen bei der Programmierung von Ereignisdiskreten Steuerungen. Hier gibt es die Möglichkeit, die Programmierung grafisch-, funktions- oder ablauforientiert vorzunehmen. Die Fachsprachen haben ihren spezifischen Gerätebezug verloren und sind inzwischen international standardisiert (IEC-Norm). Der Leistungsumfang solcher Fachsprachen reicht z. T. bis an Hochsprachen heran, so dass damit sogar intelligente Automatisierungsfunktionen problemlos programmiert werden können. Die Alternativ zu den Fachsprachen wären objektorientierte und die Echtzeit- sowie Multitaskingfähigkeit unterstützende Hochsprachen.

#### **4 Gesamteinschätzung**

Wie aus den vorstehenden Ausführungen hervorgeht, besteht ein hoher Grad an Übereinstimmungen bei der Gestaltung der Rechen- und technologischen Prozessen unter Einschluss der automatischen Steuerung mittels Prozessrechner. Daraus ergeben sich enge Beziehungen zwischen den Fachgebieten Informatik und Automatisierungstechnik, die eine Zusammenführung unter dem Dach der Prozessinformatik nahe legen. Der Prozessrechnereinsatz stellt spezifische Anforderungen, wie Echtzeitfähigkeit, Bildung von Zeitfunktionen, Interruptbehandlung, spezielle Kommunikationsgestaltung, hohe Systemverfügbarkeit u. a. Die hierzu entwickelten Lösungen können die klassische Informatik durchaus bereichern. Umgekehrt unterstützt die Informatik durch Bereitstellung von hochleistungsfähigen *Tools* und anwendungsgerichteten Fachsprachen wesentlich die Arbeit der Automatisierungsingenieure beim Systementwurf und der Erstellung von Anwendungsprogrammen.

Die Prozessinformatik überdeckt allerdings nicht das Gesamtgebiet der Automatisierungstechnik. Die Unterschiede betreffen vor allem die technische Ausführung von Automatisierungslösungen. Zu den Besonderheiten zählen vor allem die benötigten (Prozess-)Peripheriekomponenten und speziellen Ausführungen der Hardwaremodule, die den bestehenden Einsatzbedingungen genügen müssen.

## Literatur

Nehmer, J.: Softwaretechnik für verteilte Systeme. Springer-Verlag, 1985

Beuschel, J.: Prozeßsteuerungssysteme. R. Oldenbourg Verlag, 1994

Herrtwich, R. G., Hommel, G.: Kooperation und Konkurrenz. Springer-Verlag, 1989

Weller, W.: Script zur Vorlesungsreihe „Prozessrechentchnik“. Humboldt-Universität zu Berlin, Inst. f. Automatisierungstechnik (unveröffentlicht)

Weller, W.: Automatisierungstechnik im Überblick. Beuth Verlag GmbH, 2008