1
Motivation

1.1
Regelungstechnik
Dirk Abel

Nicht nur die Komplexität verfahrenstechnischer Prozesse, sondern auch die Anforderungen an Qualität, Umweltverträglichkeit und Rentabilität steigen stetig an. Aus diesen Gründen kommt dem Einsatz höherer Verfahren der Automatisierungs- und Leittechnik in der Prozessindustrie eine ständig wachsende Bedeutung zu [1]. Unter den Verfahren des sog. *Advanced Control*, d. h. den höheren Regelungsmethoden, haben dabei insbesondere modellgestützte prädiktive Regelungen, bei denen ein mathematisches Modell des dynamischen Prozessverhaltens zum integralen Bestandteil des Regelungsgesetzes wird, ein großes Verbesserungspotential und auch Praxistauglichkeit bewiesen.

In allen Industriebereichen sind die Entwickler automatisierungstechnischer Funktionen gewohnt, moderne Entwurfs- und Simulationsumgebungen wie z. B. MATLAB/Simulink [2, 3] einzusetzen. Zur schnellen, zuverlässigen und damit auch kostengünstigen Umsetzung von Automatisierungslösungen auf verschiedensten Hardwareplattformen wurden Verfahren und Werkzeuge zum *Rapid Control Prototyping* (RCP) entwickelt [4, 5]. Kennzeichnend für RCP ist eine automatische Codegenerierung ausgehend von der Simulations- und Entwicklungsumgebung, die eine direkt am realen Prozess einsetzbare Automatisierungslösung schafft. Dadurch wird ein durchgängiger Entwicklungsprozess gewährleistet, der es ermöglich, mit wenig Aufwand Erprobungen am realen Prozess durchzuführen und den Übergang von Prototyp zu Produkt gleichzeitig in Simulation und Realität zu vollziehen. Zusätzlich dazu ist die Identität der tatsächlich genutzten Softwarefunktionen mit dem Funktionsmodell in der Simulation garantiert. Weiterhin erleichtert eine übersichtliche Versionsverwaltung in Bibliotheken die Verwendung unterschiedlichster Regelungs- und Steuerungskonzepte und führt damit zu einer erhöhten Flexibilität bei der Erprobung. Eine ausführlichere Erläuterung und Betrachtung der RCP-Methodik erfolgt im Abschnitt 1.1.1.

RCP wird z. B. in der Automobilindustrie schon sehr erfolgreich eingesetzt, für Anwendungen in der Prozessleittechnik sind derartige Entwurfsverfahren jedoch

Integration von Advanced Control in der Prozessindustrie: Rapid Control Prototyping.
Herausgegeben von Dirk Abel, Ulrich Epple und Gerd-Ulrich Spohr
Copyright © 2008 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31205-4

bisher nahezu unbekannt. Der Einsatz von RCP ist auf Grund der verfügbaren gerätetechnischen Randbedingungen gegenwärtig auf eine Toolkopplung in der Ebene der Bedienstationen eines Prozessleitsystems beschränkt, z. B. als MATLAB/Simulink-OPC–Client [6]. Eine solche Gerätekonfiguration erlaubt den Einsatz von Verfahren des Advanced Control in den nicht zeitkritischen übergeordneten Automatisierungsebenen, die die Teilprozesse koordinieren. Der Zugriff auf die prozessnahe Ebene und damit die Möglichkeit, Advanced Control mit Hilfe von Methoden des RCP in die elementaren Automatisierungsfunktionen einer verfahrenstechnischen Anlage einzubringen, bleibt hingegen verschlossen.

Ein weiteres Problem besteht darin, dass der Einsatz neuer, komplexerer regelungstechnischer Verfahren im Prozessleitsystem mit gewissen Schwierigkeiten verbunden ist. Diese umfassen z. B. eine oft unzureichende kommunikations- und ablauftechnische Integration in das leittechnische Umfeld, die Problematik der organisatorischen Einbindung in die betrieblichen Engineering-, Wartungs- und Reengineeringprozesse sowie eventuell nicht beachtete Sicherheits- und Robustheitsfragen. Es wäre daher in jedem Falle wünschenswert, die bewährten Strukturen – insbesondere die sicherheitsrelevanten – soweit wie möglich beizubehalten und als Rückfallebene, die z. B. bei Störungen des Advanced-Control-Algorithmus eingreift, weiter zu verwenden.

Das in diesem Buch beschriebene Konzept “Totally Integrated Advanced Control” (TIAC) ermöglicht sowohl die Integration von Advanced Control in die prozessnahen Funktionen als auch den Aufbau einer sichereren Rückfallstrategie, die auf konventioneller Prozessregelung beruht.

1.1.1 Rapid Control Prototyping (RCP)

Rechnergestützte Entwicklungsmethodiken erhalten für Ingenieure in allen Anwendungsgebieten zunehmende Relevanz. Im Bereich der Regelungs-, Steuerungs- und Automatisierungstechnik wird Rapid Control Prototyping als Methode und Werkzeug angesehen, das einen integrierten Entwicklungsprozess erlaubt, der von der Spezifikation über die Modellbildung und Simulation des Prozesses über alle weiteren notwendigen Schritte bis hin zur Verifikation des Gesamtprozesses führt.

Gerade im Bereich der Regelungstechnik eröffnen sich dadurch zahlreiche neue Möglichkeiten. So können auch höhere Regelungsalgorithmen zur Erprobung und Umsetzung gelangen. Aufgrund der Möglichkeit, Entwicklungsprozesse wesentlich flexibler und durchgängiger zu gestalten, können auf diese Weise auch Ansätze einer dynamischen Gesamtoptimierung des zu entwickelnden Prozesses in einem sehr frühen Entwicklungsstadium eingebracht werden. Damit wird es deutlich einfacher möglich sein, regelungs- und steuerungstechnische Belange verstärkt bereits in der Entwurfsphase der Prozesse einzubringen, anstatt wie bisher häufig erst im Nachhinein hinzugezogen zu werden, um dann mit “heilenden Anforderungen” konfrontiert zu werden.

Fortschritte in der Theorie der Regelungstechnik und die stetig steigende Rechenleistung wirtschaftlich einsetzbarer Prozessoren führen zu einem Wandel sowohl im Entwicklungs- als auch im Auslegungsprozess regelungstechnischer Systeme. In klassischen Ansätzen wird das Prozessverhalten lediglich untersucht, um daraus nach häufig heuristischen Regeln die Parameter für einfache Regelalgorithmen abzuleiten. Moderne Verfahren zeichnen sich dagegen z. B. dadurch aus, dass das Wissen über die Prozesseigenschaften in unterschiedlichen Formen unmittelbar im Regler genutzt wird.

Man spricht in diesem Zusammenhang auch von "höheren" Regelungsalgorithmen, die auch für schwierigere Anwendungen, wie z. B. nichtlineare, gekoppelte Mehrgrößenregelungen, eingesetzt werden und deren Anwendung die Beherrschung mancher Prozesse unter den vorgegebenen Randbedingungen erst möglich macht. Diese Algorithmen zeichnen sich häufig durch die Einbeziehung von modellbasiertem Prozesswissen im Regler aus und erfordern deshalb andere Kompetenzen als klassische Verfahren. Auf der anderen Seite müssen die Algorithmen weiterhin so gestaltet werden, dass sie auf der eingesetzten Hardware in Echtzeit ausführbar sind. Weitere Anforderungen entstehen aus der zunehmenden Notwendigkeit, auch hybride und miteinander vernetzte Systeme behandeln zu können. Hieraus entsteht zusätzlicher Bedarf nicht nur hinsichtlich des Entwurfs solcher Systeme, sondern insbesondere auch hinsichtlich der Verifikation und Validierung.

Die praktische Nutzung dieser Verfahren bedingt einen verstärkten Einsatz von Techniken der Modellbildung und Simulation, erfordert aber auch, komplexere Algorithmen zuverlässig, flexibel und effizient umzusetzen. Es ergibt sich damit die in Abb. 1.1 schematisch dargestellte Verschiebung des Aufwandes innerhalb der Entwicklungsprozesse. Bei den klassischen Verfahren werden sehr stark vereinfachte Modelle genutzt, sodass große Teile des Entwurfs und der Parametrierung des Automatisierungskonzepts unmittelbar am realen Prozess durchgeführt und überprüft werden müssen. Dieser Vorgang ist häufig sehr zeitaufwändig und kostspielig, sodass, wenn überhaupt, nur wenige Varianten und Alternativen untersucht werden können.

Bei modernen Entwurfsmethoden verschiebt sich dieses Bild drastisch zu größerem Aufwand bei der Modellbildung und eventuell auch der damit möglichen Simulation, sodass die Inbetriebnahme am realen Prozess idealer Weise mit

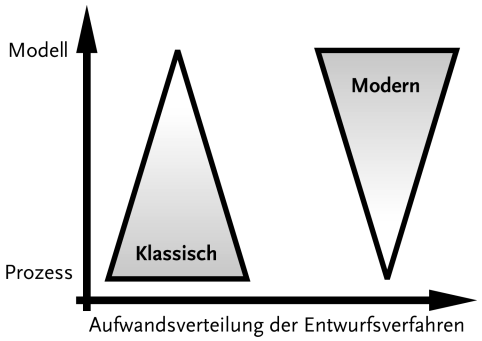


Abb. 1.1 Verschiebung des Entwicklungsaufwands.

einem in der Struktur und der Parametrierung vollständig an den realen Prozess angepassten Automatisierungskonzept erfolgen kann. Dadurch ergibt sich als zusätzlicher Vorteil die Möglichkeit, bereits in einem sehr frühen Entwicklungsstadium auch Designschwächen oder gar -fehler aufdecken und beheben zu können. Diese Entwicklung spiegelt sich unter anderem in der wachsenden Bedeutung von Simulations- und Programmierwerkzeugen in der ingenieurberuflichen Praxis wider.

Der erforderliche Brückenschlag zwischen der abstrakten Prozess- und Regelungsbeschreibung in Modellen und der von der Hardware abhängigen Realisierung ist als Methodik in der Praxis bisher wenig verbreitet. Aufgrund der Bedeutung dieser Entwicklungsprozesse haben sich zunehmend Firmen am Markt etabliert, die die Lücke durch den integrierten Einsatz ihrer Produkte zu schließen suchen [7], ohne dies bisher vollständig gewährleisten zu können. Das liegt unter anderem auch daran, dass sich notwendige Werkzeuge und Beschreibungsformen teilweise noch im Stadium der Forschung befinden. Dies betrifft z. B. formalisierbare Beschreibungsformen für gemischt kontinuierlich und diskrete Systeme und deren systematischen Steuerungsentwurf, aber auch viele andere noch zu lösende Probleme in diesem Umfeld.

Moderne Entwurfsmethoden orientieren sich sehr häufig an der Vorgehensweise nach dem sog. V-Modell (Abb. 1.2). Ein typischer Entwicklungsprozess für die Regelung eines komplexen Systems beginnt danach, ausgehend von der Aufgabenstellung und der Analyse, mit der Modellbildung, gefolgt von einem ersten Automatisierungsentwurf in der Simulation. Hier können unterschiedliche Automatisierungsstrukturen untersucht und durch Anwendung entsprechender Entwurfsmethoden eingestellt und verifiziert werden. Dies kann man als Systemsimulation bezeichnen; oftmals stellt diese das wesentliche Entwicklungswerkzeug der regelungstechnischen Arbeit dar. In dieser Umgebung kann ebenfalls sukzessive die erforderliche Rechenleistung und Genauigkeit untersucht werden, um das notwendige Hardwareprofil definieren zu können.

Sind hierdurch geeignete Algorithmen gefunden und in der Simulation erprobt worden, beginnt häufig die Phase der händischen Software-Entwicklung für die Zielhardware, also für die am realen Prozess eingesetzte Steuerung (z. B. bei

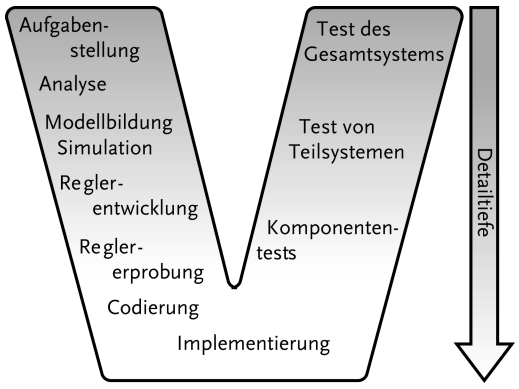


Abb. 1.2 Klassisches V-Modell für die Entwicklung einer Regelung.

fertigungs- oder verfahrenstechnischen Anlagen) oder das Steuergerät (z. B. im Automobilbereich). Ziel ist es, zunächst die gefundenen Algorithmen unter Einhaltung der Restriktionen wie Rechengenauigkeit und Echtzeit nachzubilden. Hieran anschließend findet eine Erprobung am Prozess statt; der bereits realisierte Algorithmus wird auf seine Nutzbarkeit hin getestet. In den meisten Fällen treten Abweichungen des Prozessverhaltens von dem Verhalten des in der Simulation verwendeten Modells auf, wodurch eine neue Iteration in diesem Entwicklungsprozess durch Modifikation des Modells für die Systemsimulation gestartet wird.

Die Möglichkeit bei diesem Vorgehen, von Schritt zu Schritt Iterationen durchführen zu können, hat bereits zu einer deutlichen Verbesserung im Entwicklungsprozess geführt. Die ursprüngliche klassische Reglerentwicklung wurde schließlich fast vollständig am realen Prozess durchgeführt. Die Möglichkeit zu Iterationen blieb jedoch bisher weitestgehend auf die vertikale Richtung beschränkt, eine horizontale Iteration wird erst dann eingeleitet, wenn bei den Tests am Ende des V-Modells Designfehler deutlich werden, die nur auf der Einstiegsseite des V-Modells behoben werden können.

Mittlerweile ist es möglich, weitestgehend durchgängige Toolketten zu nutzen, die von der Modellbildung bis zur Codegenerierung und -verifikation den Automatisierungstechnik-Ingenieur unterstützen. Damit wird es prinzipiell möglich, bereits in der Entwurfsphase den Gesamtprozess zu betrachten und gegebenenfalls nicht nur auf der Steuerungs- und Regelungsebene Anpassungen zu erproben, sondern auch am zu regelnden Prozess. Damit rückt die Automatisierungstechnik vom Ende des Entwicklungsprozesses, welches häufig durch "heilende" Anforderungen aufgrund von Designfehlern geprägt ist, in eine Entwicklungsphase, in der eine dynamische Gesamtoptimierung zumindest noch möglich ist.

1.1.2 HW/SW-in-the-Loop-Simulation

Diese modellbasierten Vorgehensweisen, die auch unter den Stichworten *HW-* und / oder *SW-in-the-Loop-Simulation* zu finden sind, dienen dazu, eine komfortable, bedienerfreundliche, graphisch programmierbare Simulationsoberfläche durch automatisierte Code-Generierung mit einer Zielhardware zu verbinden. Werden alle dazu notwendigen Werkzeuge in einer gemeinsamen integrierten Umgebung zusammengefasst, ergibt sich unmittelbar auch die Möglichkeit, Iterationen in der horizontalen Ebene des V-Modells vornehmen zu können.

Vorteile ergeben sich dadurch sowohl für methodische als auch für inhaltliche Aspekte. Methodisch nutzen diese Ansätze auf der abstrakten Modellebene häufig dieselben graphischen Beschreibungsformen wie in der Systemsimulation. Diese stammen meist aus dem jeweiligen Anwendungsgebiet und erleichtern damit den Transfer des Wissens der jeweiligen Experten in Modelle. Der wesentliche Schritt besteht nun in der Möglichkeit der automatischen Code-Generierung. Ist der reale Prozess verfügbar, kann z. B. für die gefundene Automatisierungsstruktur direkt ausführbarer Code erzeugt und mit Hilfe einer leistungsfähigen skalierbaren Echtzeit-Hardware mit Prozessankopplung eine Erprobung durchgeführt werden.

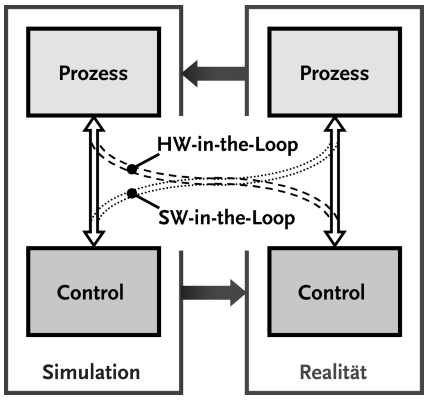


Abb. 1.3 Strukturen des Rapid Control Prototyping.

Diesen Schritt, bei dem der Prozess real und die Regelung auf einer leistungsfähigen Entwicklungsplattform "simuliert" wird, kann als "Software-in-the-loop" bezeichnet werden. Iterationen lassen sich auf diese Art und Weise zwar nicht umgehen, aber durch die Beschleunigung der einzelnen Zyklen und die Entlastung des Ingenieurs bei der Implementierung ist die Entwicklung schneller und kostengünstiger durchführbar. Analog ist auch die Codegenerierung des Prozessmodells möglich – hierdurch eröffnen sich zwei wesentliche Möglichkeiten.

Die erste ähnelt der Systemsimulation, bei der sowohl der Prozess als auch die Automatisierungslösung simuliert wird, findet jedoch unter Echtzeitbedingungen auf zwei Plattformen statt. Hierbei ist man nicht an die Verfügbarkeit und die Beschränkungen der realen Prozesse gebunden. So lassen sich z. B. auch sicherheitskritische Untersuchungen durchführen und solche zu Komponenten, die real (noch) nicht verfügbar sind. Die zweite Möglichkeit ist in Abb. 1.3 als "Hardware-in-the-loop" bezeichnet und erleichtert die Erprobung und Freigabe. Ist ein auf der Zielhardware lauffähiger Algorithmus generiert worden, kann mit Hilfe der Echtzeit-Prozesssimulation über entsprechende Versuche eine risikoarme Verifikation von allen auftretenden sicherheitskritischen Prozesszuständen simuliert werden.

Auf der Seite der Codegenerierung bestehen die methodischen Vorteile darin, dass das Wissen über die Umsetzung formaler (graphischer) Beschreibungen in effizient ausführbaren Code an zentraler Stelle gesammelt und verwaltet werden kann und Anforderungen wie z. B. die Fehlerfreiheit und Reproduzierbarkeit einfacher zu erfüllen sind, als bei händischer Umsetzung. Die wesentlichen inhaltlichen Vorteile bestehen darin, dass der jeweilige Entwickler sich ausschließlich auf seine Kernkompetenzen konzentrieren kann. Der Prozessingenieur bekommt idealer Weise eine Umgebung zur Verfügung gestellt, deren Gestaltung sich an den in seinem Fachgebiet üblichen Darstellungen orientiert, um die notwendigen Modelle zu entwickeln. Der Automatisierungstechniker nutzt diese Modelle für den Steuerungs- und Regelungsentwurf. Dafür werden die Modelle in derselben Umgebung zur Verfügung gestellt, aber Werkzeuge und Darstellung an seine Belange angepasst. Gleiches vollzieht sich mit den weiteren an der Entwicklung beteiligten Experten. Die Verwaltung von Modellen und zugehörigen bzw.

abgeleiteten Regelungsstrukturen und deren Dokumentation wird damit ebenfalls leichter möglich, da jeder Experte in seiner gewohnten Umgebung arbeiten kann, ohne dass die zugehörigen Informationen in unterschiedlichen Programmen abgelegt werden müssen.

1.2 Leittechnik

Ulrich Epple

1.2.1 Entwicklungsebenen der Leittechnik

Die Analyse der Entwicklung in der Leittechnik kann auf drei Betrachtungsebenen erfolgen: der *Basissystemebene*, der *leittechnischen Systemdiensteebene* und der *Anwendungsebene*. In Abb. 1.4 sind die drei Ebenen mit ihren Funktionalitäten dargestellt.

Die unterste Ebene ist die *Basissystemebene*. Ihr werden z. B. die Rechner (als Hardwarekomponenten), das Betriebssystem, die Bussysteme, die verwendeten Datenbanken und die Grafiksysteme zugerechnet. Die Basissystemebene bildet die Entwicklungsplattform, auf der ein Hersteller sein Leitsystem aufbaut.

Die zweite Ebene ist die *leittechnische Systemdiensteebene*. Sie wird durch den Hersteller implementiert. Durch ihre Funktionalität wird aus einem allgemeinen verteilten Rechnersystem ein Prozessleitsystem. Die Funktionalität dieser Ebene unterstützt die Realisierung der leittechnischen Konzepte und erlaubt eine einfache Konfiguration der Anwenderfunktionalität. Sie garantiert bestimmte Sys-

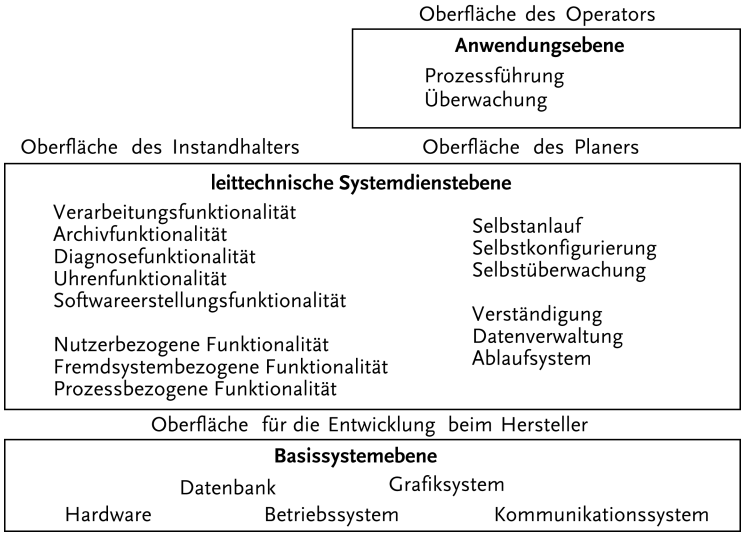


Abb. 1.4 Systemebenen der Prozessleitsysteme (nach einem Vorschlag des DKE K930 von 1992).

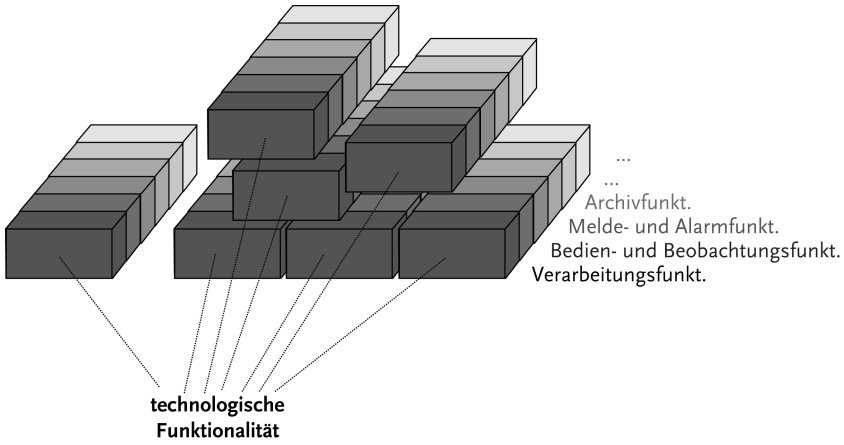


Abb. 1.5 Gliederung der Anwendungsfunktionalität nach technologischen und leittechnischen Gesichtpunkten.

temeigenschaften wie z. B. Verfügbarkeit, Fehlerarmut, Betriebssicherheit, Echtzeitverhalten usw. und unterstützt ein einfaches Handling in Planung und Betrieb. Die von dieser Ebene angebotene Funktionalität ist die Plattform, auf der der Planer seine spezielle Anwendungsfunktionalität implementiert. Im Idealfall verbirgt die leittechnische Systemdiensteebene die Besonderheiten der Basissystemebene gegenüber dem Planer vollständig. Inhaltlich umfassen die leittechnischen Systemdienste sowohl Rahmenfunktionalitäten als auch fertige Funktionspakete. Eine Rahmenfunktionalität wie z. B. das Funktionsbausteinsystem bietet dem Planer eine Sprachumgebung, in der er die Anwendungsfunktionalität einfach und fehlerarm formulieren kann (konfigurieren statt programmieren). Bestimmte leittechnische Systemfunktionen werden in allen Anwendungen in gleicher Weise benötigt. Diese werden als fertige Funktionspakete vom Leitsystem zur Verfügung gestellt. Sie müssen im Einzelfall nur noch parametriert werden. Dazu gehören z. B. das Meldesystem, das Archiv, die Systemdiagnose, aber auch speziellere Pakete wie z. B. das Rezeptsystem.

Die dritte Ebene ist die *Anwendungsebene*. Sie entsteht durch die Projektierung und umfasst die zur Lösung der speziellen Aufgabenstellung erforderliche Anwendungsfunktionalität. Man kann die Awendungsfunktionalität zunächst nach zwei Kriterien gliedern: nach der Struktur der technologischen Aufgabenstellung und nach der leittechnischen Art der Funktionalität. Der erste Gliederungsansatz führt auf eine komponentenorientierte Struktur. Zu jeder technologischen Aufgabenstellung gibt es eine leittechnische Funktionseinheit, die alle zur Lösung dieser technologischen Aufgabenstellung erforderlichen Elemente umfasst. Grundlage des zweiten Gliederungsansatzes ist eine Unterscheidung verschiedener leittechnischer Funktionsarten. Man unterscheidet z. B. die Verarbeitungsfunktionalität, die Bedien- und Beobachtungsfunktionalität, die Meldefunktionalität, die Archivfunktionalität usw. Insgesamt ergibt sich die in Abb. 1.5 dargestellte Gliederung.

Die Anwendungsfunktionalität jedes Leitsystems ist nach diesem Matrixmuster gegliedert. Die Weiterentwicklung der leittechnischen Konzepte erfolgt vorwiegend auf der Grundlage des komponentenbasierten Ansatzes. Er erlaubt die einfache und geschlossene Hantierung von technologischen Einheiten. Die Realisierung der leittechnischen Systemfunktionen erfolgt teilweise in den Komponenten. Voraussetzung ist ein zwischen den Komponenten abgestimmtes Standardkonzept der leittechnischen Systemfunktionen.

1.2.2 Entwicklungstendenzen Basissystemebene

Unter dem Druck der Entwicklung in der Informationstechnik hat sich in den letzten Jahren im Bereich der Basissysteme ein tiefgreifender Wandel vollzogen. Der Übergang von speziellen leittechnischen Lösungen auf allgemeine Betriebssystem-, Kommunikations-, Datenhaltungs- und Präsentationsstandards hat sowohl technologisch als auch unternehmenspolitisch zu völlig neuen Strukturen geführt. Technologisch wurden die Systeme durch die Ankopplung an die allgemeine Entwicklung in der IT-Welt signifikant leistungsfähiger und kostengünstiger. Dafür handelte sich die Leittechnik mit diesen nicht für ihre Anforderungen konzipierten Systemen eine Vielzahl von technologischen Problemen ein, z. B. bezüglich Verlässlichkeit, Investitionssicherheit, Pflegbarkeit und einfacher Handhabbarkeit [10, 11, 17]. In der Zwischenzeit haben die Hersteller Wege gefunden diese Probleme zu lösen oder zumindest geeignet mit ihnen umzugehen. Politisch hat die Übernahme von allgemeinen Standards für die Leittechnik zu einer neuen Situation geführt. Die Weiterentwicklung der Konzepte und Strukturen der Basissysteme werden nun nicht mehr von den Anforderungen der Leittechnik getrieben, sondern von den angebotenen Lösungen aus der allgemeinen IT-Technologie.

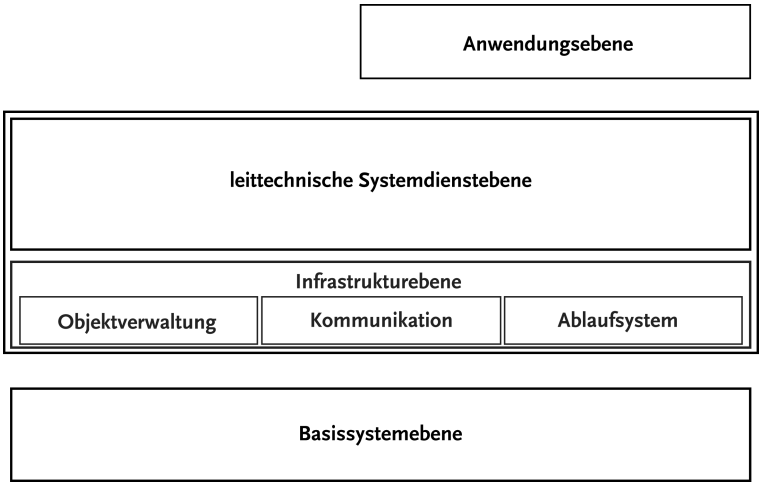


Abb. 1.6 Infrastrukturebene zur Sicherung der leittechnischen Basis-Systemanforderungen.

Im Ergebnis führt dies zu einer Situation, in der die Komponenten und Lösungskonzepte des Basissystems in kürzesten Entwicklungszyklen innoviert und restrukturiert werden müssen, ohne dass dafür aus leittechnischer Sicht eine Notwendigkeit besteht. Ziel der weiteren Entwicklung muss es daher sein, leittechnische Infrastrukturebenen zu bilden, die einerseits robust gegen Änderungen der Basistechnologien sind, andererseits aber auch einfach und effektiv auf die Basistechnologien abgebildet werden können. In Abb. 1.6 ist dargestellt, wie der Einsatz einer solchen Infrastrukturebene zwischen dem Basissystem und den leittechnischen Systemdiensten gedacht ist.

Funktional muss die Infrastrukturebene mindestens eine Sprach-und Ausführungsplattform für die Objektverwaltung, die Basiskommunikation und das Ablaufsystem bereitstellen. Das System ACPLT [8, 12] zeigt in einem Referenzmodell, wie eine solche Infrastruktur konzeptionell und realisierungstechnisch aufgebaut sein könnte. Ideal wäre mittelfristig eine Abbildung von allen für die Leittechnik relevanten Grundfunktionen des Basissystems in der Infrastrukturebene.

1.2.3 Entwicklungstendenzen Anwendungsebene

In den letzten Jahren und Jahrzehnten hat sich an der implementierten Funktionalität der Anwendungsebene wenig geändert. Füllstands-oder Durchflussregelkreise, Pumpensteuerungen, Grenzwertüberwachungen und ähnliche Standardaufgaben werden heute noch wie vor 20 Jahren mit den gleichen einfachen Standardalgorithmen gelöst. Neue Funktionalitäten sind nicht oder nur vereinzelt und punktuell hinzugekommen. Nach der Einführung der Rezeptsysteme in den 80er Jahren [14] hat bis heute kein neues Konzept mehr den Einzug in die betriebliche Praxis geschafft. In der Zwischenzeit müssen jedoch eine Reihe dieser bewährten Konzepte wie z. B. das Zusammenspiel zwischen Bediener, Prozess und Anlage, die Archivierung, die Melde-und Alarmfunktionalität usw. sowohl konzeptionell als auch lösungstechnisch als veraltet angesehen werden. Trotz erheblicher Beharrungskräfte bei den Anwendern, den Engineeringfirmen und den Herstellern wird sich ein Konzeptwechsel nicht mehr lange aufhalten lassen. In den nächsten Jahren ist eine grundlegende Neukonzeption des Aufbaus der Anwenderfunktionalität zu erwarten.

Hintergrund ist der Druck auf die Engineeringkosten einerseits und die Anforderung nach einer signifikanten Erweiterung der Funktionalität andererseits. Aus der Weiterentwicklung der Methoden und aus dem Bestreben nach einer Optimierung und Integration der Informationsströme in den Produktionsbetrieben haben sich in den letzten Jahren neue Aufgabenstellungen für die Prozessleittechnik entwickelt. Diese müssen durch entsprechende Funktionalitäten auf der Systemseite unterstützt werden. Wie in Abb. 1.7 dargestellt gehören dazu z. B. das *Supply Chain Management* [18], das *Asset Management* [13, 15], die Realisierung einer durchgängigen *Life Cycle Dokumentation*,das *Performance Monitoring* [9] und Funktionalitäten zur Unterstützung des *eCommerce*.

Diese funktionalen Anforderungen der Betriebsleit- und Produktionsleitebene an die Prozessleitebene bringen eine Reihe von Problemen mit sich. Die gravierendsten sind:

- Signifikante Zunahme der implementierten Funktionalität. (Projektier-und Pflegekosten, Übersichtlichkeit);
- Zusammenführen der Informationshaushalte (Modellanpassung, breitbandige und flexible Schnittstellen);
- Zusammenspiel mit fremden Entwicklungs-und Ablaufumgebungen
- Nutzung der leittechnischen Systemdienste für Funktionen der Betriebsleit- und Produktionsleitebene;
- Abgrenzung der Funktionalität und Verantwortlichkeiten.

Der Grundgedanke der klassischen Prozessleitsysteme und ihrer leittechnischen Systemdienste ist die integrierte Struktur, das heißt, der Hersteller sorgt dafür, dass alle Funktionen zueinander passen und sich gegenseitig unterstützen. Ein solches Konzept ist im größeren Rahmen der Betriebs- und Produktionsleitebene nicht umzusetzen. Zunächst ist es unwahrscheinlich, dass alle Systeme von einem Hersteller (und dann auch noch aus einer Systemlinie) kommen und selbst wenn dies so wäre, würde es die Integrationsfähigkeit des Herstellers überfordern und seine Innovationskraft lähmen. Man muss in dem betrieblichen IT-Verbund also mit heterogenen Systemarchitekturen leben. Der Festlegung einer normierten Grundarchitektur und von bestimmten Schnittstellen als stabilen Entwicklungsknoten kommt daher eine entscheidende Bedeutung zu.

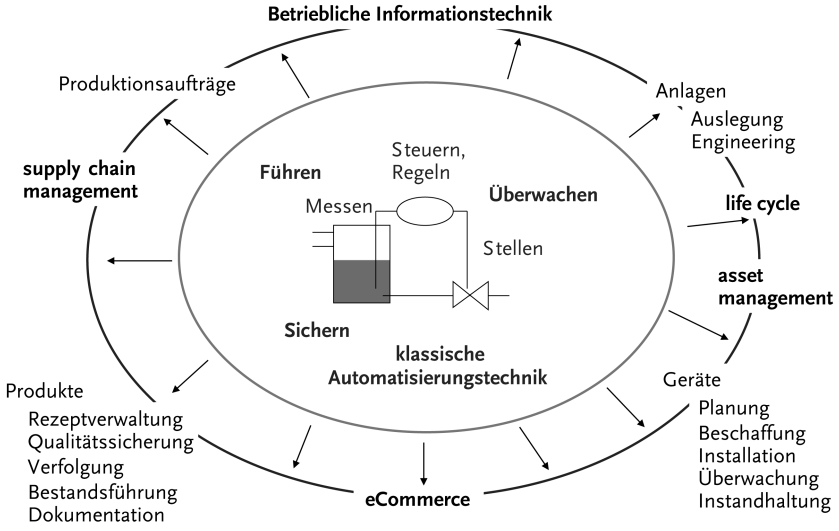


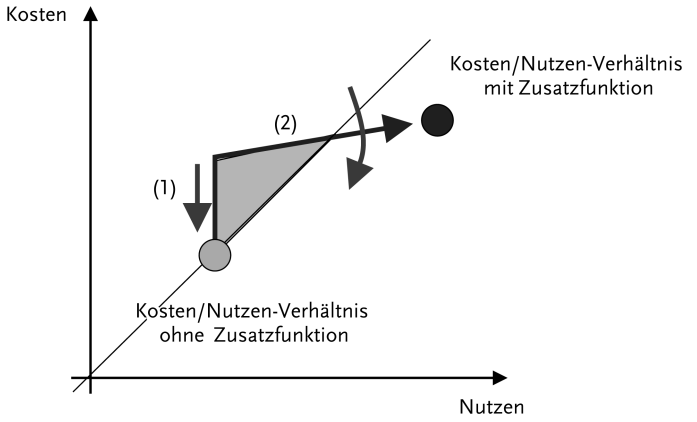
Abb. 1.7 Prozessleitsysteme als integrale Bestandteile der betrieblichen Informationstechnik.

1.2.4 Zusatzfunktionen

Unter einer *Zusatzfunktion* versteht man, wie der Name schon andeutet, eine Funktion, die zusätzlich zu den Auslegungsfunktionen realisiert wird. Sie ist für den normalen Auslegungsbetrieb nicht unbedingt erforderlich. Zusatzfunktionen dienen der Optimierung der Produktionsprozesse und Betriebsabläufe. Sie unterstützen z. B. die Anlagenwirtschaft, die Instandhaltung, die Produktionsdokumentation oder die Prozessoptimierung durch entsprechende IT-Funktionalität. Per Definition besitzen sie damit nur mittelbaren Einfluss auf die Produktion. Ein Ausfall einer Zusatzfunktion kann zwar zu einem suboptimalen Betrieb, jedoch nicht zu einer Störung der Produktion führen.

Durch die "Zusatz"-eigenschaft ergeben sich einige charakteristische Merkmale von Zusatzfunktionen.

- Zusatzfunktionen unterliegen einer strikten Kosten-Nutzen-Betrachtung.
  Dadurch, dass Zusatzfunktionen im Gegensatz zu den Schutz- und Auslegungsfunktionen für die Durchführung des Produktionsprozesses nicht direkt und unbedingt erforderlich sind, unterliegen sie einer strikten Kosten-Nutzen-Betrachtung. Bei einer einzelnen Schutzfunktion lässt sich nicht diskutieren, ob auf sie aus Kostengründen verzichtet werden kann, bei einer Zusatzfunktion sehr wohl. Erschwerend kommt hinzu, dass der Nutzen einer einzelnen Zusatzfunktion in vielen Fällen gering ist und sich oft nur schwierig quantitativ belegen lässt. So hat z. B. eine Auswertung in den Mitgliedsfirmen der NAMUR ergeben, dass der Nutzen, der sich mit Funktionen des *Geräte Asset Management* erzielen lässt, zwar nachweisbar ist, sich gegen den Aufwand jedoch nur schwer rechnet [16, 19]. Aus dieser Sicht heraus ist es nicht verwunderlich, dass sich die Anwender bei der Einführung entsprechender Funktionspakete zurückhalten, insbesondere, da die Erstinstallationskosten und die Instandhaltungskosten solcher Zusatzfunktionspakete typischerweise nicht unerheblich sind. Das Ergebnis einer solchen Analyse sollte allerdings nicht sein, sich mit dieser Situation abzufinden und auf den Einsatz von Zusatzfunktionen zu verzichten. Wie in Abb. 1.8 skizziert sollte vielmehr nach Konzepten gesucht werden, mit denen die Kosten sowohl für die Erstinstallation als auch für den funktionsmengenabhängigen Betreuungsaufwand drastisch gesenkt werden können. Eine drastische Senkung der Kosten kann nur gelingen, wenn die Zusatzfunktionen mit auf sie zugeschnittenen Werkzeugen effektiv geplant und betreut werden können.



(1) Erstinstallation
(2) Implementierungskosten je Einzelfall

Abb. 1.8 Kosten-Nutzen-Betrachtung zum Einsatz einer Zusatzfunktion.

Eine weitestgehende Automatisierung ihrer Handhabung ist für einen breiten durchgängigen Einsatz unverzichtbar.

- Zusatzfunktionen benötigen eine spezielle Entwicklungs-und Ablaufumgebung.
  Zusatzfunktionen bauen typischerweise auf komplexen Methoden und anspruchsvollen numerischen und informationstechnischen Lösungskonzepten auf. In vielen Fällen sind die klassischen leittechnischen Beschreibungssprachen und Entwicklungstools nicht geeignet. Für einen effektiven Einsatz ist die Verwendung von für die entsprechende Aufgabenstellung speziell entwickelten Tools unerlässlich. Diese Tools erzeugen Ergebnisse die wiederum eine spezielle Ablaufumgebung verlangen.
- Integration von externen Lösungen
  Zusatzfunktionen gehören typischerweise zu Paketen, die nicht alle aus der Hand eines Leitsystemlieferanten kommen. In den meisten Fällen wird man sogar verschiedene Zusatzpakete von verschiedenen Herstellern beziehen. Die Integration in die leittechnische Umgebung erfordert strikte systemtechnische Regeln, die auch vertragsrechtlich eindeutige Verhältnisse schaffen.
- Entwurf und Betreuung durch Spezialisten
  Der Entwurf und die Betreuung von Zusatzfunktionen erfordert typischerweise spezielles Wissen und spezielle Kenntnisse. Dazu gehört auch ein geübter Umgang mit den verwendeten Tools. In vielen Fällen ist das Hin-

zuziehen von Spezialisten für die Implementierung und Betreuung erforderlich.

- Temporäre Realisierung
  Zusatzfunktionen werden nicht unbedingt schon während der Planung mit den Auslegungsfunktionen fest implementiert, sondern erst später bedarfsweise oder situationsabhängig hinzugefügt. Sie müssen damit während des Betriebs flexibel und dynamisch handhabbar sein.

1.2.5 Entwicklungstendenzen im Bereich der leittechnischen Systemdienste

Im Gegensatz zum Basissystem hat sich an der Funktionalität des leittechnischen Betriebssystems in den letzten Jahren nicht viel geändert. Ziel der Entwicklung war es vielmehr, die bestehenden leittechnischen Funktionalitäten und Systemeigenschaften trotz der gravierenden Änderungen in der Systembasis zu erhalten. Darüber hinaus kam von Anwendungsseite wenig Druck die leittechnische Systemfunktionalität zu innovieren. Bisher kamen die Anwender mit der klassischen leittechnischen Funktionalität gut zurecht und standen Erweiterungen und konzeptionellen Innovationen eher skeptisch gegenüber. Die Hauptanforderung aus Anwendersicht war die Restabilisierung des leittechnischen Betriebssystems nach den Umwälzungen auf der Basissystemebene. Dies wird sich jedoch in Zukunft ändern. Die beschriebenen neuen Anwendungsfunktionen werden für eine optimierte Anlagen- und Prozessführung zunehmend unerlässlich. Sie können jedoch nur dann kostengünstig und fehlersicher implementiert werden, wenn die leittechnischen Systemdienste dazu die entsprechende Unterstützung bieten. Grundsätzlich sind die leittechnischen Systemdienste nicht nur für die Prozessleitsysteme relevant, sondern für alle Systeme, insbesondere auch der Betriebsleit- und Produktionsleitebene. Es muss also Ziel sein, z. B. für Funktionen wie Melden, Alarmieren, Archivieren, Anzeigen usw. einen einheitlichen generischen Kern zu entwickeln und allgemein zu normen. Dies ist ein heres und fernes Ziel. Für die Migration ist es von besonderem Interesse ein Nebeneinander von hochverfügbaren, getesteten und unveränderlichen Auslegungsfunktionen und neuen, frei handhabbaren flexiblen Zusatzfunktionen zu organisieren. Die leittechnischen Systemdienste müssen dafür den Rahmen zur Verfügung stellen. Das Konzept der *gesicherten Funktionsebenen* bietet dazu einen Ansatz.

1.2.6 Gesicherte Funktionsebenen

Das Konzept der gesicherten Funktionsebenen (*Functional Integrity Level*, FIL) wird in Abb. 1.9 erläutert. Grundlage ist eine Hierarchisierung der Automatisierungsfunktionalität auf drei Ebenen.

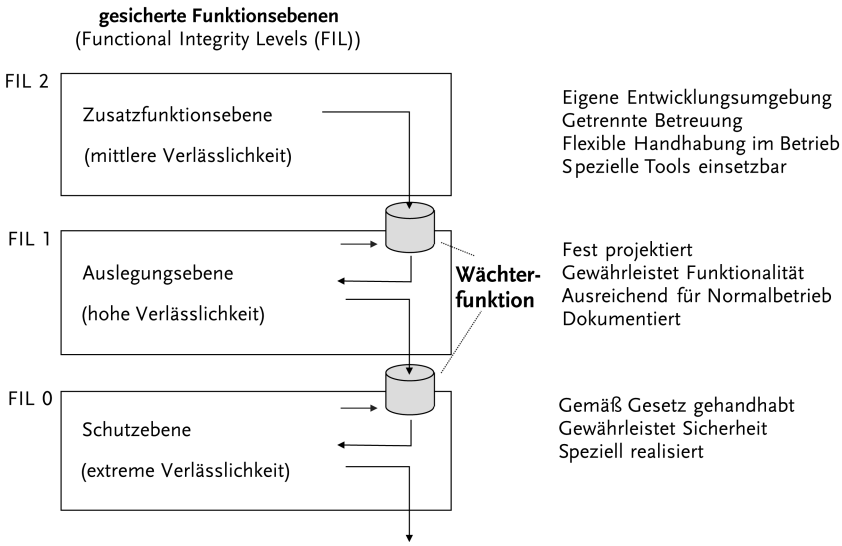


Abb. 1.9 Konzept der gesicherten Funktionsebenen (FIL).

Schutzebene

Die unterste Ebene ist die Schutzebene. Sie enthält die mit PLT-Mitteln realisierten Schutzfunktionen. Die Funktionalität ist fest spezifiziert, geprüft und abgenommen. Sie ist Grundlage der Betriebsgenehmigung der Anlage. Sie kann nur nach einem aufwändigen Genehmigungsprozess verändert werden. An die Funktionserbringung werden extreme Anforderungen z. B. bezüglich der Verlässlichkeit gestellt, die im Allgemeinen nur durch eine spezielle technologische Realisierung (hardwaretechnisch fixierte Realisierung, spezielle sicherheitsgerichtete SPS, spezielle Kommunikationssysteme ...) und Handhabungsregeln erfüllt werden können. Es gehört zu den elementaren Regeln des Systemdesigns, dass die Schutzfunktionalität weder durch die Auslegungsfunktionalität selbst noch durch Fehler in deren Handhabung in irgend einer Weise beeinträchtigt werden darf. Diese *Rückwirkungsfreiheit* wird durch das leittechnische Realisierungskonzept der Schutzebene sichergestellt.

Auslegungsebene

Die nächste Ebene ist die Auslegungsebene. Sie enthält die Auslegungsfunktionen. Sie ist fest projektiert, dokumentiert, Grundlage der Gewährleistung und stellt mit hoher Verlässlichkeit die für die operative Prozessführung und Prozess- und Anlagenüberwachung erforderliche Funktionalität zur Verfügung. Die leittechnischen Systemdienste unterstützen die Projektierung, Verwaltung und sichere Ausführung der Auslegungsfunktionen. Dazu gehört neben der Bereitstellung entsprechender Sprachmittel und Systemdienste auch die Zurverfügungstellung einer entsprechenden Systemarchitektur. In Abb. 1.10 ist z. B. der klassische Aufbau einer Automatisierungskomponente dargestellt.

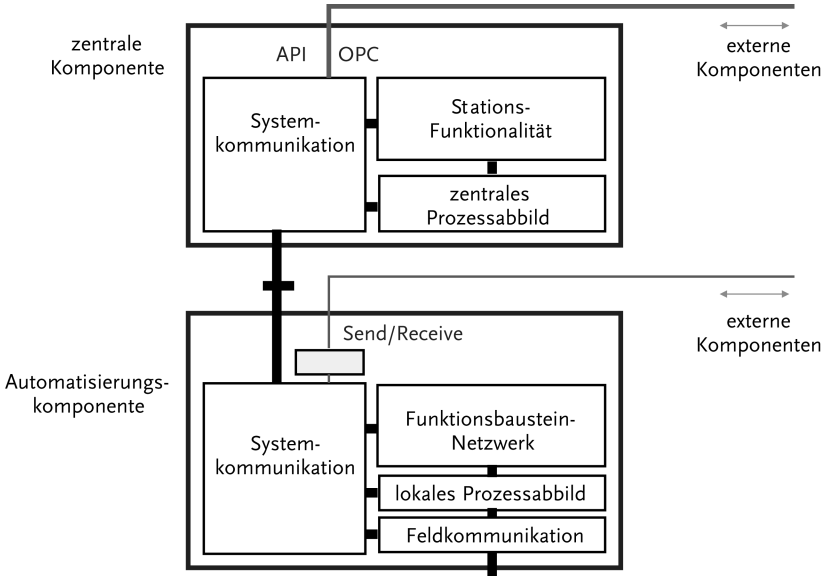


Abb. 1.10 Standardarchitektur einer Automatisierungskomponente.

Die Auslegungsfunktionalität ist in einem fest projektierten Funktionsbausteinnetzwerk hinterlegt. Die Bausteine haben über das lokale Prozessabbild Zugriff auf die aktuellen Mess-und Stellwerte in den Feldgeräten. Die Feldkommunikation erfordert eventuell einige Konfigurationsvorgänge, wird ansonsten jedoch vom Leitsystem systemseitig realisiert. Das gleiche gilt für die Systemkommunikation. Sie organisiert und sichert den gesamten Informationsaustausch mit den Leitfunktionen in den zentralen Komponenten (Bedienen und Beobachten, Archivieren, Melden ...). Die Systemdienste der Auslegungsebene unterstützen die Implementierung und Ausführung von technologischen Funktionseinheiten (siehe Abb. 1.5) mit allen leittechnischen Aspekten. Sie garantieren die korrekte Ausführung der Funktionsbausteinbearbeitung und die deterministische Bereitstellung der erforderlichen Systemressourcen. Sie bieten jedoch keinerlei Hilfestellung zur Regelung der Interaktion zwischen technologischen Funktionseinheiten.

Zusatzfunktionsebene

Für die Realisierung von Zusatzfunktionen bieten sich zwei Wege an: die Realisierung in einem Fremdsystem mit Ankopplung über das API der zentralen Komponente oder Send/Receive-Blöcke der Automatisierungskomponente oder die Realisierung in speziellen Funktionsbausteintypen und Integration in die Auslegungsfunktionalität. Beide Wege besitzen erhebliche Nachteile. Ein Grundproblem der Handhabung von Zusatzfunktionen ist das Fehlen einer Regelung, wie technologische Funktionseinheiten untereinander interagieren dürfen, also z. B. wie eine Zusatzfunktion auf eine Auslegungsfunktion einwirken darf. Das

hier vorgeschlagene Lösungskonzept ist die in Abb. 1.9 dargestellte Einführung der Zusatzfunktionsebene als dritter Funktionsebene mit einer definierten Schnittstelle zur Auslegungsebene. Der Kern des Konzepts ist die Wächterfunktionalität zwischen der Auslegungsfunktionsebene und der Zusatzfunktionsebene. In dieser Wächterfunktionalität sind die Interaktions- und Eingriffsmöglichkeiten, die die Auslegungsebene zulässt, explizit festgelegt. Die Wächterfunktionalität wird durch entsprechende Systemdienste auf der Auslegungsebene realisiert. Die Auslegungsebene sichert sich durch diese Funktionalität selbst. Unter diesen Randbedingungen können die Zusatzfunktionen beliebig realisiert und frei gehandhabt werden. Insbesondere können fremde Tools verwendet, dynamische Rekonfigurationen vorgenommen und auch Lösungen mit nur mittlerer Verlässlichkeit eingesetzt werden.

1.2.7 Wächterfunktionalität für die Auslegungsebene

Während die Wächterfunktionalitäten auf der Schutzebene einer relativ einfachen Logik folgen und einfachen Randbedinungen genügen, sind bei den Wächterfunktionalitäten der Auslegungsebene komplexe und von der Art der Interaktion abhängige Mechanismen erforderlich. Der Aufbau eines Regelwerks steht hier erst am Anfang. Zur Verdeutlichung sollen drei Situationen, die im Themenbereich dieses Buches eine besondere Rolle spielen, diskutiert werden.

Rückwirkungsfreies Auslesen von Struktur-und Zustandsinformationen
Der einfachste Fall ist die Anforderung Struktur-und Zustandsinformationen aus der Auslegungsebene für die Nutzung in der Zusatzfunktionsebene auszulesen. In diesem Fall ist die Interaktionsfunktion in sich selbst rückwirkungsfrei. Die Wächterfunktionalitäten müssen lediglich dafür sorgen, dass der Auslesevorgang nicht zu einer unzulässigen Belastung der Ressourcen führt. Der Vorgang ist allerdings nur dann wirklich rückwirkungsfrei, wenn das Auslesen ohne zusätzliche Projektierung auf seiten der Auslegungsebene erfolgen kann. Das heißt, eine Zusatzfunktion kann dynamisch frei vorgeben, welche Information sie wann lesen möchte und die Wächterfunktionalität navigiert die Anfrage an die richtige Instanz und organisiert den Transfer. Prinzipiell stehen alle Informationen der Auslegungsebene einer solchen Abfrage über die Wächterfunktionalität offen zur Verfügung. Im Einzelfall kann der Zugriff durch die Security-Funktionen verhindert werden. Die Integration der Security-Funktionen in das Wächtermodell ist jedoch eine eigene Thematik, die hier nicht betrachtet werden soll.

Vorgabe von Parametern und Zustandswerten
Wesentlich kritischer ist die Realisierung eines Ebenenschutzes, wenn über die Zusatzfunktionen eine gezielte Einwirkung auf die Prozessführung und letztendlich den Prozess ausgelöst werden soll. Dies geschieht z. B. bei Optimierungsfunktionen, Advanced-Control-Anwendungen, Parametrierungen oder auch im Grenzfall bei der Implementierung übergeordneter Fahrhilfen. In diesen Fällen müssen

Sollwerte, Stellwerte oder Parameter der Auslegungsfunktionen von den Zusatzfunktionen verstellt werden. Der Schreibzugriff ist durch die Wächterfunktionalität zu kontrollieren. Dabei können den Schreibzugriffen eine Reihe von Einschränkungen auferlegt werden:

- Ist der Wert prinzipiell durch Zusatzfunktionen modifizierbar;
- Ist in der aktuellen Situation eine Wertänderung erlaubt;
- Liegt der Vorgabewert in einem zulässigen Wertebereich;
- Wird eine maximale Änderungsgeschwindigkeit nicht überschritten;
- ...

Alle für die Prüfungen benötigten Parameter und Methoden sind Teil der Auslegungsfunktion. Um für die Verwaltung und Prüfung einen einheitlichen Rahmen zu schaffen, bieten sich mehrere Lösungen an. In diesem Buch werden der Einsatz einer Auftragsschnittstelle und die Verwendung eines standardisierten Fahrweisenrahmens als Lösungen vorgeschlagen.

Auftragsschnittstelle
Die Auftragsschnittstelle ist eine spezielle Diensteschnittstelle. Über sie können einer Auslegungsfunktion Aufträge erteilt werden. Die Vorgabe und Inhalte der Aufträge werden von der Zielfunktion selbst auf ihre Zulässigkeit überwacht. Über die Auftragsschnittstelle können z. B. Parameter verstellt, Fahrweisen vorgewählt oder auch Führungswerte eingestellt werden. In Kapitel 4 ist das Konzept beschrieben.

Komponentenrahmen
Eine noch tiefere Integration erlaubt das Komponentenkonzept. In diesem Fall stellt die Auslegungsebene eine Kapsel zur Verfügung, der eine beliebige externe

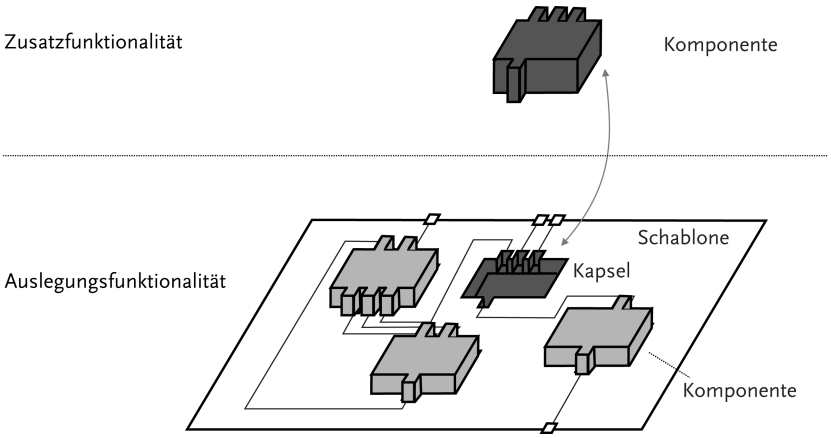


Abb. 1.11 Kapselung von Zusatzfunktionen

Ausführungskomponente zugeordnet werden kann. Abb. 1.11 verdeutlicht die Situation.

Die Auslegungsebene ist als Schablone realisiert. Die Kapseln sind nicht nur Gliederungskonstrukte, sondern bilden *reale Rahmenwerke*, die als Interfaces den Informationsaustausch mit den eingebetteten Realisierungskomponenten implementieren. Im Allgemeinen sind die Komponenten feste Funktionalitäten der Auslegungsebene. Spezielle erweiterte Kapseln lassen jedoch auch die Einbettung von externen Realisierungskomponenten aus der Zusatzfunktionsebene zu. In diesem Fall enthält das Interface komplexe Prüf-und Überwachungsroutinen zur Realisierung der Wächterfunktionalität. Um die Rückwirkungsfreiheit auch wirklich sicherstellen zu können, wird der gesamte Informationsaustausch an der Schnittstelle von der Kapsel aus ausgeführt. Im TIAC-Projekt wird dieses Konzept zur Integration einer beliebigen Advanced-Control-Lösung in einen PID-Reglerrahmen genutzt. Darüber hinaus bieten sich jedoch vielfältige Möglichkeiten zur *sicheren Integration* von *unsicherer Funktionalität*.

1.3 Leitsysteme

Gerd-Ulrich Spohr

1.3.1 APC-Anwendungen in Prozessleitsystemen

1.3.1.1 Historische Entwicklung der Prozessleitsysteme

Um den Einsatz höherer Regelungstechnik auf Prozessleitsystemen beurteilen zu können, ist zunächst ein kurzer Rückblick auf die Entstehungsgeschichte dieser Systeme erforderlich. Dezentrale Prozessleitsysteme, kurz DCS genannt, wurden erstmals um 1979 / 1980 in den Markt eingeführt, wobei die Firmen Honeywell und Yokogawa darum wetteifern, wer nun das erste "echte" DCS vorgestellt hat.

In den Zeiten davor wurden Anlagen in pneumatischer und / oder elektrischer Einzelgerätetechnik ausgerüstet und automatisiert. Bei der Einzelgerätetechnik wird jede benötigte Funktion, z. B. eine Messung, Regelung oder Ein-/Aus-Funktion, durch ein dediziertes einzelnes Gerät oder eine Kombination einzelner Geräte realisiert. Diese Geräte wurden in der Regel in der Messwarte auf großen Tafeln montiert, da die Bedienoberfläche, z. B. die Anzeige oder der Schalter, fester Bestandteil des Gerätes war. Die Anordnung der Geräte entsprach dem Prozessfluss, welcher oberhalb der Bedien- und Anzeigetafeln in Form eines großen R&I-Schemas mit integrierten Rückmelde-Lampen dokumentiert wurde. Der Vorteil der Einzelgerätetechnik lag auf Grund der weitestgehenden Verteilung der Aufgaben in der hohen Robustheit gegenüber Einzelfehlern. Dem standen allerdings eine ganze Reihe von Nachteilen gegenüber. Bauart bedingt benötigte diese Aufbautechnik erheblichen Platz in der Messwarte und auf alten Bildern sind oft Messwartenfronten von mehr als 20 m Länge zu bewundern. Allein diese Ausdehnung machte einen gewissen Mindestbestand an Personal zur Bedienung in

der Messwarte erforderlich. Ein weiterer Nachteil war die Inflexibilität gegenüber Änderungen der Prozess-Struktur. Signalwege waren fest vorgegeben, zusätzliche Signalverknüpfungen mussten explizit verdrahtet werden und somit waren schnelle Strukturumschaltungen praktisch kaum zu realisieren. Daher war diese Art der Automatisierung am ehesten noch für Conti-Anlagen mit langen, konstanten Produktlaufzeiten geeignet und hat sich dort auch am längsten gehalten.

Um eine höhere Flexibilität zu erreichen, wurden etwa ab dem Beginn der 70er Jahre sog. Prozessrechner für die Automatisierung von Anlagen eingesetzt. Es handelte sich dabei um Computer, die von ihrem Aufbau und den Anforderungen an die Umgebungsbedingungen für den Einsatz in der Industrie geeignet waren. Natürlich benötigten sie klimatisierte Schalträume und besonders geschultes Bedienpersonal, aber im Vergleich zu den damals üblichen Computern in Rechenzentren waren diese Systeme bemerkenswert robust und universell einsetzbar. Die bekanntesten Vertreter dieser Rechnergeneration sind die Typen PDP11 und die VAX750 der Firma Digital Equipment, die noch bis in die 90er Jahre von verschiedenen Herstellern für die Prozessautomatisierung eingesetzt wurden.

Bei den Prozessrechnern wurden alle relevanten Prozess-Signale auf entsprechenden Eingangskarten aufbereitet und digitalisiert. Diese Daten wurden dann dem Rechner zur Verarbeitung übergeben. Ausgangsdaten wurden entsprechend in elektrische Signale gewandelt und den Aktoren im Feld zugeführt. Die Verarbeitung der Daten auf dem Prozessrechner erfolgte anhand von Programmen, die individuell für den jeweiligen Prozess entworfen und erstellt wurden. Die Programmierung wurde zunächst aus Zeit- und Platz-Gründen in Assembler vorgenommen, denn damalige Rechner besaßen Speicherplatz in der Größenordnung einiger zig Kbyte. Mit zunehmender Speichergröße kamen dann Hochsprachen wie FORTRAN oder das besser strukturierte PASCAL zum Einsatz. Da die Programme individuell auf den jeweiligen Prozess und die dort zu lösenden Automatisierungs-/ Regelungs-Probleme zugeschnitten wurden, kann man davon ausgehen, dass in diesen Systemen auch erstmals höhere Regelalgorithmen zum Einsatz kamen. Diese mussten jedoch die begrenzten Ressourcen der zur Verfügung stehenden Rechner berücksichtigen.

Den Vorteilen wie hohe Flexibilität und freie Programmierbarkeit standen jedoch auch bei diesem Ansatz einige gravierende Nachteile gegenüber. Durch die zentrale Bearbeitung aller Informationen stieg das Risiko eines Totalausfalls durch eine Rechnerstörung, so dass in der Regel die Rechner grundsätzlich redundant ausgelegt wurden, was entsprechende Kosten nach sich zog. Des Weiteren war zum Betrieb und insbesondere zur Störungssuche am Rechner entsprechend geschultes Fachpersonal erforderlich, was einen zusätzlichen Kostenfaktor und einen erheblichen Abstimmungs- und Kommunikationsaufwand bedeutete.

Das Ziel der weiteren Entwicklung war es, die Robustheit und Transparenz der dezentralen Informationsverarbeitung der Einzelgerätetechnik mit den Vorteilen des Prozessrechneransatzes zu verbinden. Dies wurde mit den Fortschritten der Mikroprozessortechnik möglich. Auf der Basis von Mikroprozessoren konnten leistungsfähige und dabei doch preiswerte und robuste Rechner aufgebaut werden, die dann dank der inzwischen entwickelten Kommunikationstechnik auf der

Basis von Bussystemen zu Rechnerverbünden zusammengeschaltet werden konnten. Ein dezentrales Prozessleitsystem bestand somit aus einer Reihe von Mikrocomputern, die dezentral jeweils eine Teilaufgabe bearbeiten konnten und über das gemeinsame Bussystem untereinander Informationen austauschen und so wie ein großes Gesamtsystem agieren konnten. Mit diesem Ansatz begann die Ära der Prozessleitsysteme die bis heute anhält und in verschiedenen Ausprägungen am Markt verfügbar ist.

1.3.1.2 Technische Realisierung von APC-Anwendungen

Bei der Entwicklung der dezentralen Prozessleitsysteme wurde die bei den Prozessrechnern übliche individuelle Programmierung auf der Basis von Hochsprachen durch die viel leichter erlernbare Funktionsbausteintechnik ersetzt. Dabei werden komplexe Programme aus einzelnen Programm-Modulen, den sog. Funktionsbausteinen, zusammengesetzt. Diese Funktionsbausteine sind Bestandteil der Firmware eines Leitsystems und werden vom Hersteller mit ausgeliefert. Diese Funktionsbausteine decken die üblicherweise zur Automatisierung einer Anlage benötigten Funktionen weitgehend ab und werden meist in Form einer Bibliothek wieder verwendbarer Softwaremodule realisiert. Da mit den Funktionsbausteinen ein möglichst breites Anwendungsgebiet abgedeckt werden soll, sind die Funktionen mehr oder weniger standardisiert, z. B. Standard-PID-Algorithmus oder Standard-Motor- bzw. -Ventilansteuerung. Spezielle Ausprägungen, wie z. B. Fuzzy-Control oder Mehrgrößen-Regler, sind daher selten anzutreffen.

Wie lassen sich dann trotzdem solche höherwertigen Aufgaben bewältigen. Dazu bieten sich zwei verschiedene Möglichkeiten an.

Viele Leitsystem-Hersteller bieten den Anwendern die Möglichkeit, eigene individuelle Funktionsbausteine mit Hilfe von Hochsprachen zu erstellen und diese dann wie die Standardbausteine zu benutzen und so das Leistungsspektrum des Systems zu erweitern. Bei der Erstellung individueller Funktionsbausteine sind jedoch einige Randbedingungen zu beachten. Auch wenn die Leistungsfähigkeit heutiger Prozessleitsysteme um mehrere Größenordnungen über denen früherer Prozessrechner liegt, so sind deren Ressourcen wie Speicherplatz und insbesondere Bearbeitungszeit nicht unbegrenzt verfügbar. Bei der Realisierung individueller Funktionsbausteine tut man also gut daran, den dafür erforderlichen Ressourceneinsatz kritisch zu betrachten und unter Kosten-/Nutzen-Betrachtungen zu analysieren. Denn die Ressourcen auf einer prozessnahen Komponente eines Leitsystems sind in der Regel die kostbarsten und der Problemstellung oft nicht angemessen.

Hier bietet sich dann die zweite Möglichkeit an: die Auslagerung der Bearbeitung höherer Funktionen auf einen eigenen Rechner. Bei der Auslagerung der Bearbeitung steht dem Anwender für die Realisierung der Funktion die volle Bandbreite der heute verfügbaren Werkzeuge zur Verfügung. Ressourcen sind auf einem eigenständigen Rechner im Überfluss vorhanden. Somit entpuppt sich die Kommunikation und Koordination mit dem Leitsystem als das eigentlich zu lösende Problem.

Der Zugriff auf den Informationshaushalt eines Prozessleitsystems erfolgt in der Regel über die Bedien- und Beobachtungs-Komponente. Diese verfügt über den Gesamtbestand aller relevanten Informationen und stellt in der Regel auch eine, oft proprietäre, Schnittstelle zur Verfügung. Da der Informationshaushalt einer HMI-Komponente auf die Bedürfnisse eines Operators optimiert wurde, stehen die Daten mit Refresh-Zyklen von 1–2 s zur Verfügung. Der für die ausgelagerte Bearbeitung gewählte Rechner wird daher an die Schnittstelle der HMI-Komponente angeschlossen und es erfolgt ein zyklischer Datenaustausch der relevanten Informationen. Typischerweise kann von einer erreichbaren Zykluszeit von 2–6 s ausgegangen werden. Mit der Verbreitung der standardisierten OPC-Schnittstelle ist der Aufwand für die Realisierung der Kommunikation drastisch gesunken, die erreichbaren Zykluszeiten liegen aber nach wie vor oft oberhalb von 1–2 s.

Da die höherwertigen Automatisierungsaufgaben in der Prozessindustrie oft in deutlich längeren Zeitintervallen ablaufen, ist dies in vielen Fällen keine Einschränkung und daher ist die Auslagerung höherwertiger Aufgaben auf eigene Rechner eine oft geübte Praxis bei der Lösung komplexer Automatisierungsaufgaben.

Es bleibt jedoch bei einem solchen Systemverbund das Problem der Sicherheit und der Koordination, insbesondere bei einem derart "lose" gekoppelten System. Was passiert bei einem Ausfall des überlagerten Systems und wie kann man eventuell "auseinander" gelaufene Systeme wieder so synchronisieren, dass anschließend ein stoßfreier Übergang in einen geführten Betrieb möglich ist. Hier kommen die Überlegungen zur "Wächter"-Funktion (siehe Abschnitt 1.2) zum Zuge. Durch entsprechende Überwachungs- und Synchronisations-Funktionen muss sichergestellt werden, dass in jedem beliebigen Betriebszustand nur die jeweils zulässigen Übergänge freigeschaltet und vom Anwender genutzt werden können. Dieser Problematik haben wir uns im TIAC-Projekt gestellt und entsprechende Lösungen erarbeitet.

In der bisherigen Praxis hat man sich der Problematik oft dadurch entzogen, dass die Systeme definitiv entkoppelt waren und die Kopplung durch eine externe und verantwortliche Instanz, den Operator, vollzogen wurde. In diesen Fällen wurden vom überlagerten System Parameter und Sollwerte vorgeschlagen und konnten vom Operator, nach Prüfung der Plausibilität und Zulässigkeit, in das unterlagerte System übernommen werden.

Im Rahmen des TIAC-Projektes haben wir den Versuch unternommen, die Vorteile einer direkten Implementierung im Leitsystem mit den Freiheitsgraden der Nutzung eines eigenständigen externen Computers zu verbinden. Durch die direkte Kopplung der Ein-/Ausgangssignale über den Feld-Bus, im vorliegenden Fall den PROFIBUS DP, können Zykluszeiten bis hinunter in den ms-Bereich realisiert werden. Durch den Einsatz gekoppelter Zustandsautomaten konnte die schon angesprochene "Wächter"-Funktionalität realisiert werden, die eine sichere und bedienerfreundliche Nutzung komplexer Algorithmen ermöglicht. Durch die Nutzung eines eigenen dedizierten Computers stehen für die Realisierung der APC-Funktionen Ressourcen zur Verfügung, die die der leistungsfähigsten Pro-

zessrechner vergangener Tage um mehrere Größenordnungen übertreffen. Gleichzeitig können zur Entwicklung der Algorithmen die modernsten Werkzeuge wie MATLAB und Simulink eingesetzt werden, die ihrerseits wiederum eine umfangreiche Bibliothek hochwertiger und erprobter Lösungen für eine Vielzahl von Aufgaben bereitstellen.

1.3.1.3 APC-Anwendungstypen

Was sind denn nun typische APC-Anwendungen, die in der Prozessindustrie und somit im Umfeld von Prozessleitsystemen realisiert werden.

Dabei macht es Sinn die Anwendungen anhand der Automatisierungspyramide und den darin beschriebenen Ebenen zu gliedern.

Die oft standortübergreifenden Optimierungsaufgaben der Unternehmensleitebene werden in der Regel in unternehmensweiten ERP-Systemen realisiert, z. B. mit SAP oder ORACLE, um nur zwei der wichtigsten Vertreter dieser Systemgruppe zu nennen.

In der Betriebsleitebene sind die MES-Systeme (*Management Execution System*) anzutreffen. Diese bauen auf dem Informationshaushalt der Prozessleitsysteme auf und dienen oft auch als Bindeglied zu den überlagerten, unternehmensweiten ERP-Systemen. Gleichzeitig werden auch eine ganze Reihe von Verwaltungs- und Optimierungsaufgaben in den MES-Systemen bearbeitet. Insbesondere die gesamte Anlagenlogistik, die Betriebsmittelverwaltung, die Optimierung der Anlagenauslastung und die Produktverfolgung sind typische Leistungen solcher Systeme. Weiterhin werden anlagenspezifische Kennwerte, z. B. KPI (*Key Performance Index*), ermittelt und der Produktionsleitung zur Verfügung gestellt.

Tabelle 1.1

Ebene	Automatisierungsfunktion	Zeithorizont
Unternehmens-Leitebene	Supply Chain Management, längerfristige Produktionsplanung, Kostenanalyse, andere betriebswirtschaftlich orientierte Funktionen	Tage ... Monate
Betriebsleitebene	Online-Prozessoptimierung mit theoretischen Prozessmodellen, statische Arbeitspunktoptimierung und Trajektorien-Optimierung, erweiterte Protokollierung und Betriebsdaten-Archivierung (Prozess- und Labor-Informationssysteme), Process Performance Monitoring, kurzfristige Produktionsplanung, Rezeptur-Erstellung und -Verwaltung	Stunden ... Tage

Ebene	Automatisierungsfunktion	Zeithorizont
Prozessleitebene	Bedienen und Beobachten, Registrieren/Protokollieren, *Model Predictive Control, Fuzzy-Logik und Fuzzy-Control, andere moderne Regelungsverfahren (adaptive Regelungen, robuste Regelungen, nichtlineare Regelungen, Internal Model Control, Zustandsregelung, Entkopplungsregelungen ...), Softsensoren (Beobachter, Kalman-Filter, neuronale Netze, Regressionsmodelle) Prozessdiagnose, Control Performance Monitoring (CPM),* Statistical Process Control (SPC), Rezeptur- und Ablaufsteuerungen	s ... min
Prozessnahe Ebene	PID-Basisregelungen und *vermaschte Regelungen* (Kaskadenregelung, Verhältnisregelung, Störgrößenaufschaltung, Split-Range, Override-Regelung, *Gain Scheduling, Smith-Prädiktor, ...*), *Selbsteinstellung* von PID-Reglern, Schutz- und Verriegelungsfunktionen	ms ... s
Feldebene	Sensoren und Aktoren	ms

In der Prozessleitebene und prozessnahen Ebene finden wir dann die Funktionen, die oft unter dem Schlagwort *Advanced Process Control* zusammengefasst werden. Neben den verschiedenen Ausprägungen höherer Regelungstechnik sind aber auch Softsensor-Funktionen sowie erweiterte Diagnose und Anlagenzustands Überwachungen wichtige und oft rechenintensive Funktionen, die bei der Realisierung eine genauere Betrachtung des Ressourcenbedarfs erfordern.

Hier bietet sich das TIAC-Konzept als eine mögliche Realisierungsvariante an. Das Konzept ist so ausgelegt, dass es sich auch nachträglich leicht und rückwirkungsfrei in eine bereits bestehende Anlage integrieren lässt. Der TIAC-Rechner wird lediglich als zusätzlicher Teilnehmer am PROFIBUS angemeldet und über diesen mit den relevanten Informationen aus dem Datenhaushalt des Leitsystems versorgt. Durch den entsprechenden Funktionsbaustein des Leitsystems wird die sichere und rückwirkungsfreie Kommunikation mit der APC-Anwendung sichergestellt und gleichzeitig Mechanismen für die Synchronisation der Zustandsautomaten bereitgestellt. Auf dem TIAC-Rechner können nun unter Verwendung des Bausteinrahmens und der Zustandsautomaten beliebig komplexe APC-Anwendungen realisiert werden. Die in unserem Projekt realisierten Reglerapplikationen sollen da lediglich als Beispiel und “proof of concept” dienen.

Typischerweise werden Anlagen zunächst einmal ganz konventionell geplant und unter Verwendung von Standard-Funktionen (z. B. PID) “zum Laufen” gebracht. Nach einiger Zeit im laufenden Betrieb stellt sich dann die Frage, ob man nicht mit der ein oder anderen höherwertigen Regelungsfunktion oder einer modellgestützten Fahrweise den Wirkungsgrad der Anlage verbessern könnte. Allerdings wird kein Produktionsleiter gravierende Änderungen an seiner laufenden Anlage zulassen.

Hier kommt nun das TIAC-Konzept ins Spiel. Die Anlage bleibt wie sie ist, lediglich ein zusätzlicher Teilnehmer am Feldbus kommt hinzu. Damit wird auf dem TIAC-Rechner eine sichere "Spielwiese" für die Verfahrens- und Regelungs-Ingenieure eröffnet, auf der sie in Ruhe und unter Zugriff auf die Betriebsdaten ihre APC-Anwendung entwickeln und testen können. Auch der Übergang in den "scharfen" Betrieb ist recht unproblematisch, da ja jederzeit ein Rückfall auf den bisherigen, bewährten Zustand gegeben ist. Eine Tatsache, die von den meisten Produktionsleitern als Vorbedingung für eine Erprobung gefordert wird.

Wir sehen daher das TIAC-Konzept als einen Beitrag dazu, den Einsatz höherwertiger Automatisierungskonzepte in der Prozessindustrie zu erleichtern und ihre Erprobung ohne Rückwirkung auf die bestehenden Anlagen zu ermöglichen.

Literatur

[1] Müller, J.: *Regeln mit SIMATIC Praxisbuch für Regelungen mit SIMATIC S7 und SIMATIC PCS7*. Siemens Publicis Corporate Publishing, Erlangen, 2. Auflage 2002.

[2] P. Krause et al.: *"Test und Bewertung von regelungstechnischen CAE-Programmen."*, atp 44 (2002), 9, Oldenbourg Verlag.

[3] Angermann et al.: *MATLAB-/Simulink-Stateflow. Grundlagen Toolboxen, Beispiele*. Oldenbourg Verlag, München 2002.

[4] Schloßer, A., Bollig, A., Abel, D.: *Rapid Control Prototyping in der Lehre*. at – Automatisierungstechnik 52 (2004), 2, Oldenbourg Verlag.

[5] Abel, D., Bollig, A.: *Rapid Control Prototyping – Methoden und Anwendungen*. Springer Verlag, Berlin 2006.

[6] Pfeiffer, B. M., Bergold, S.: *Advanced Process Control mit dem Prozessleitsystem SIMATIC PCS7*. atp – Automatisierungstechnische Praxis 44 (2002), 2.

[7] H. Hanselmann: *Beschleunigte Mechatronik-Entwicklung durch Rapid Control Prototyping und Hardware-in-the-Loop-Simulation*. at – Automatisierungstechnik 46 (1998), 3, Oldenbourg Verlag.

[8] Albrecht, H.: *On Meta-Modeling for Communication in Operational Process Control Engineering*. 2003.

[9] Dünnebier, G., vom Felde, M.: *Performance Monitoring – Ein entscheidender Beitrag zur Optimierung der Betriebsführung*. Chem. Ing. Tech., 75:528–533, 2003

[10] Hauff, T.: *Prozessleitsysteme – vom Investitionsgut zum Wegwerfartikel*. atp Automatisierungstechnische Praxis, 4:32–39, 2000.

[11] Kopec, H., Maier, U.: *Kritische Anmerkungen zu heutigen PLS*. atp Automatisierungstechnische Praxis, 3:24–28, 2005.

[12] Meyer, D.: *Objektverwaltungskonzept für die operative Prozessleittechnik*. 2002.

[13] NAMUR: *Anforderungen an Systeme für anlagennahes Asset Management*. NAMUR, NE091 Auflage, 2001.

[14] NAMUR: *Anforderungen an Systeme zur Rezeptfahrweise*. NAMUR, NE033 Auflage, 1992

[15] Nicklaus,E., Fuss, H. P.: *Online-Asset-Management*. atp Automatisierungstechnische Praxis" 5:30–39, 2000.

[16] Schneider, H. J., Trilling, U.: *Selbstüberwachung und Diagnose von Feldgeräten Teil 3: Die Umfrage der NAMUR*. atp Automatisierungstechnische Praxis, 1::49–52, 2001.

[17] Tauchnitz, Th.: *Die neuen Prozessleitsysteme – wohin geht die Reise*. atp Automatisierungstechnische Praxis, 11:12–23, 1996.

[18] Tiemeyer, E.: *Supply Chain Management – Konzeptentwicklung und Softwareauswahl in der Praxis*. atp Automtisierungstechnische Praxis, 10:24–32, 2000.

[19] Trilling, U., Stieler, S.: *Selbstüberwachung und Diagnose von Feldgeräten Teil 1: Wirtschaftliche Bedeutung für den Anwender*. atp Automatisierungstechnische Praxis, 11:42–45, 2001.