

AUTOMATIONSSOFTWARE PARALLEL ENTWICKELN

Die modulare Maschinenentwicklung

VON PETER KEMPTNER

Im Maschinenbau kommen oft noch klassische sequenzielle Methoden bei der Entwicklung der Software zum Einsatz. Ein Grund, die klassische Sichtweise zu überdenken, sind durch Standardisierung getriebene, modulare Vorgehensweisen, die auch in der mechanischen Produktentwicklung Einzug halten. B&R unterstützt Maschinenbauer mit Automation Studio 4 dabei, Entwicklungsaufgaben konsequent zu modularisieren und so zunehmende Komplexität, niedrige Engineering-Kosten und eine schnellere Markteinführung zusammenzubringen.

Im Jahr 1992 hat B&R mit Automation Studio erstmals ein integriertes Entwicklungswerkzeug für die industrielle Automatisierung eingeführt, das die Mauern zwischen den Disziplinen der Softwareentwicklung abgebaut hat. Egal ob die Ablaufsteuerung, die Bedienung oder die Visualisierung, die Antriebe oder die Sicherheitstechnik: Die Programme für die Automatisierungslösungen von B&R werden seitdem innerhalb einer durchgängigen Entwicklungsumgebung erstellt.

In den vergangenen zwei Jahrzehnten haben sich die Ansprüche an Maschinen und Anlagen deutlich gewandelt. Der Markt fordert heute auf den Nutzer zugeschnittene Maschinen hoher Komplexität, Flexibilität und Optionsvielfalt sowie die Möglichkeit zur Integration in Gesamtanlagen. Zum Preis und mit der Stabilität eines Großserienprodukts, kurzfristig verfügbar, ohne lange Inbetriebnahmezeiten und selbstverständlich allen Normen und Dokumentationsvorschriften entsprechend, so soll

die moderne Maschine beschaffen sein. Zudem möchten Käufer auch später auf sich schnell verändernde Bedarfe reagieren können und bei Nach- oder Umrüstung eigene Veränderungen der Maschinen-Software vornehmen.

Mechanische Modularisierung zum Vorbild

Um die gestiegene Produktkomplexität schnell und günstig zu beherrschen, bedienen sich die Maschinen-Konstrukteure der Standardisierung und Modularisierung. Sie schaffen Normteile und Einheitsbaugruppen für einzelne Funktionen, die mittels definierter Anschlüsse in unterschiedlichen Kombinationen zur Gesamtmaschine oder -anlage zusammengestellt werden können.

In vielen Konstruktionen gibt es eine Arbeitsteilung zwischen Spezialisten in unterschiedlichen Unterbereichen des Engineering. „Was in der mechanischen Entwicklung längst selbstverständlich ist, sollte in der Softwareentwicklung ebenso

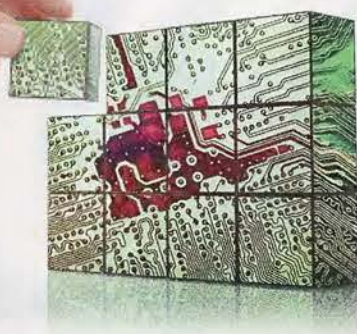
einfach möglich sein“, sagt Dr. Hans Egermeier, Business Unit Manager Automation Software bei B&R. „In der IT-Welt ist das auch längst gängige Praxis, lediglich im Maschinenbau halten sich hartnäckig Methoden aus der Frühzeit der SPS-Programmierung. Sie machen die Softwareentwicklung für komplexe mechatronische Systeme zur fast unlösbaren Aufgabe.“ Das läge nicht zuletzt an der weit verbreiteten Vorstellung von Produktionsvorgängen als sequenzielle Vorgänge. Dabei werden die zunehmend parallel stattfindenden Abläufe innerhalb von Maschinen und Anlagen immer komplexer.

Die Themen SPS, CNC, Robotik, intelligente Achsregelung, Visualisierung und Kommunikation vermaschen sich immer weiter miteinander. Um die zunehmende Komplexität funktionaler Abläufe, die Regelalgorithmen und das Steuerungsverhalten der Maschinen auch weiterhin zu beherrschen, muss auch die Software modular aufgebaut sein. In Analogie zur Mechanik mit ihren Normteilen funktioniert das über Baukästen in Form von Bibliotheken mit einzelnen Funktionen, Abläufen und Reglern.

Konsequenter modularisieren

Einzelne Unterprogramme als Funktionsblöcke in Bibliotheken abzuspeichern und durch Aufruf im Hauptprogramm zur Wirkung zu bringen, existiert als Möglichkeit bereits seit längerer Zeit und wird von Pro-

Bild: © iTestro - Fotolia.com



Wiederverwendbare Module sind wesentlich für ein intelligentes Engineering. Der Maschinenbauer profitiert von einer parallelen Entwicklung und verringerten Entwicklungszeiten und -risiken.



Dr. Hans Egermeier, Leiter des Geschäftsbereichs
Automation Software bei B&R:

„Die Software für komplexe Maschinen und Anlagen aus unabhängigen Applikationsmodulen zusammenzustellen, hilft dem Maschinenentwickler, mit der steigenden Komplexität umzugehen.“

grammierern schon fleißig genutzt. Ebenso ist in bereits existierenden Versionen von Automation Studio die Möglichkeit zur objektorientierten Programmierung in C++ vollständig integriert.

Bei dieser Art der Softwareerstellung bedient sich der Entwickler einer Struktur aus Programmklassen, die als funktionale Behälter für kleine und kleinste Funktionsprogramme dienen. Durch Zuweisung von Werten werden sie zu Programmobjekten, also Bausteinen, die aber auch ineinander geschachtelt werden können.

„Diese objektorientierte Programmierung ist die Schlüsseltechnologie zur Modularisierung von Software“, sagt Wolfgang Portugaller, Leiter Systemarchitekturen bei B&R. „Sie hilft, größere Programme übersichtlich zu halten und dadurch ihre Wartbarkeit nachhaltig zu erhöhen.“ Allerdings sei die Granularität trotz der Möglichkeiten zur hierarchischen Verkettung von Objekten noch recht fein. Trotz objektorientierter Programmierung müssen die Softwareentwickler noch alle Teile einer Maschinenprogrammierung in einem Gesamtprogramm zusammenführen, um sie als Ganzes für die Maschine oder Anlage zu übersetzen, zu testen und in Betrieb zu nehmen.

Eine der wesentlichsten Neuerungen von Automation Studio 4 ist die Modularisierung auf einer höheren Ebene durch autonom lauffähige Applikationsmodule. Diese können unterschiedlich groß sein und einzelne Funktionen, aber auch ganze Maschinenteile oder Teilmaschinen repräsentieren. In ihrem Inneren können sie hierarchisch aus einzelnen Funktionsblöcken, ganzen Programmen oder beliebigen Mischungen davon bestehen.

Entwicklungsaufgaben verteilen

Neben einer erleichterten Abbildung modularer Maschinenkonzepte erlaubt die Modularisierung mittels Applikationsmodulen eine Verteilung der gesamten Ent-

wicklungsaufgabe auf mehrere Entwickler. Nicht notwendigerweise müssen sie im selben Haus sitzen. So ist es zum Beispiel ohne großen Aufwand möglich, externe Dienstleister in die Automatisierung einzubeziehen oder Teile der Programmierung vom Kunden vornehmen zu lassen. Dabei können nicht nur funktional unterschiedliche Teilprogramme auf die jeweiligen Spezialisten aufgeteilt, sondern auch große Einzelaufgaben von mehreren Entwicklern parallel bearbeitet werden.

Das beschleunigt die Softwareentwicklung erheblich: Auf Grundlage vereinbarter Schnittstellen lassen sich Applikationsmodule gleichzeitig entwickeln und durch Simulation ausführlich testen. Darüber hinaus sind eine ganze Reihe vorgefertigter Bibliotheken und Funktionsmodule verfügbar – etwa die Steuerung von Antriebsachsen. Diese Bibliotheken und Module müssen nur noch in das eigene Projekt integriert werden.

Der Austausch von Daten zwischen den einzelnen Applikationsmodulen erfolgt mithilfe des Mappings von Prozessvariablen – ein bereits heute bewährter Mechanismus. Die Variablen müssen nicht global von außen definiert werden, denn innerhalb des Applikationsmoduls wird festgelegt, welches andere Modul zu welchen Bereichen des eigenen Adressraums Zugriff erhält. Auf diese Weise muss nicht im ersten Projektierungsschritt bereits an alles gedacht sein. Auch im Laufe der Entwicklung kann die Definition von Schnittstellen zum Datenaustausch erfolgen.

Das Kompilieren der Applikationsmodule erfolgt einzeln. Sie können daher für Tests und sukzessive in die Ziel-Hardware geladen werden, was die Fehlersuche und -behebung stark erleichtert und beschleunigt. Als Gemeinsamkeit ist ledig-

lich eine Software- und eine Hardware-Konfiguration als Information über die Laufzeitumgebung erforderlich, in der die weitgehend voneinander getrennt entwickelten Komponenten am Ende zusammenarbeiten müssen.

Änderungen erfolgen sowohl in der Prototypenphase als auch im Fall späterer Weiterentwicklungen in klar umrissenen Teilen der Gesamtanlage, was das Risiko von Qualitätsproblemen durch schnelle Änderungen minimiert. Auch kann der Zertifizierungsaufwand nach einer Änderung gering gehalten werden, da nur die von der Änderung direkt betroffenen Module einer erneuten Prüfung unterzogen werden müssen.

Auf Modul- statt Maschinenebene

Daraus ergibt sich für Maschinenbau-Unternehmen eine weitere Möglichkeit zur wirtschaftlicheren Gestaltung der Softwareentwicklung: Die bislang verbreitete Entwicklung ganzer Maschinengenerationen lässt sich durch eine sukzessive Weiterentwicklung der einzelnen funktionalen Teile ersetzen. Das sorgt für eine kontinuierliche Auslastung der Entwicklungsteams und für eine Verminderung des Drucks, der durch notorisch knappe Fertigstellungstermine entsteht.

„Tendenziell steigt auch die Softwarequalität“, kommentiert Portugaller. „In früheren Projekten fertig entwickelte Applikationsmodule sparen Zeit im ak-



Wolfgang Portugaller, Leiter
Systemarchitektur bei B&R:

„Dank objektorientierter Programmierung bleiben auch größere Programme schlank und übersichtlich, sie sind dadurch besser wartbar.“

tuellem Maschinenprojekt. Diese Zeit kann in umfangreichere Tests investiert werden, was wiederum die Inbetriebnahmezeiten verkürzt.“ Das sei nicht nur aus Sicht optimaler Nachhaltigkeit und maximaler Entwicklungseffizienz sinnvoll, sondern vor allem im Interesse des Kunden.

jbi ■

Peter Kemptner ist freier Journalist in Salzburg.