

Tschüss Spaghetti-Code

Warum eine intuitive Softwarearchitektur ein Muss für Entwickler ist



Die Komplexität neuer Anwendungen im Maschinenbau stellt hohe Anforderungen an Software-Entwickler. Diesen Anforderungen können nur Maschinenbauer gerecht werden, die sich von traditionellen, sequentiellen Methoden in der Softwareentwicklung verabschieden, und statt wild drauflos zu programmieren, Funktionsarchitektur betreiben.

Das Rollenbild des Automatisierungs-Softwareentwicklers hat sich gewandelt. „Früher waren Entwickler mit der Übersetzung von Ablauf-Anforderungen in Programmierungen beschäftigt – heute sind sie dank objektorientierter Programmierung und integrierten Entwicklungsumgebungen wahre Funktionsarchitekten in der Gestaltung mechatronischer Produkte“, sagt Franz Aschl, Innovationsmanagement bei Sigmatek.

Rückblick: Schon im klassischen Altertum wurden recht komplexe Maschinen hergestellt. Ob Flaschenzüge und Rollenbatterien zur Erleichterung von Transportaufgaben oder Katapulte als Waffensysteme. Sie alle hatten eine Gemeinsamkeit: Diese Vorrichtungen zur Kraftübertragung waren rein mechanisch aufgebaut und erfüllten eine exakt umrissene Funktion.

Von reiner Funktion zur Automation

Zu Automaten werden Maschinen durch die zusätzliche Dimension des zeitlichen Ablaufs. Dessen Steuerung beschäftigt die Ingenieure ebenfalls schon sehr lang. Viele Technologien wurden dazu angewendet,

doch erst die Elektronik machte die Steuerung auch komplexerer Abläufe nach logischen Mustern möglich. Den Durchbruch brachte die speicherprogrammierbare Steuerung. Erstmals 1968 bei GM erdacht, sollte die SPS schnell und kostengünstig zu entwickeln, die Programmierung leicht zu modifizieren sein. Im Vergleich zur zuvor eingesetzten Technik mit Relais oder hart verdrahteter Logik war sie das auch. Was sich

Die eindimensional ablauforientierte Methodik ist eigentlich unnatürlich

nicht änderte, war die Denkweise der Automatisierer. Sie orientierte sich an der Logik-Verschaltung bei der Programmierung mittels Kontaktplan (KOP) oder an der sequentiellen Arbeitsweise der Steuerrechner bei Programmierung per Anweisungsliste (AWL). Auch die Verwendung von Hochsprachen wie C änderte nichts an dieser eindimensional ablauforientierten Methodik. Diese ist eigentlich unnatürlich, weil sie kein direktes Abbild der Funktion der Maschine



Lasal bietet eine durchgängige Entwicklungsumgebung über alle Unterdisziplinen hinweg

oder des Maschinenteils darstellt. Sie macht eine Übersetzung erforderlich zwischen der Logik der mechanischen Zusammenhänge und der Logik der Befehlsabarbeitung in der Steuerungselektronik.

Mit Objekten zurück zur Funktion

Bereits vor etwa 20 Jahren erfolgte daher in einigen Branchen die Abkehr von strikt sequenzieller Programmierung. Sie wurde durch den objektorientierten Ansatz ersetzt, bei dem jede einzelne Funktion in einem als Klasse bezeichneten handlichen Block gekapselt ist, sodass eine ungewollte Beeinflussung von außen ausgeschlossen werden kann. In ihrem Inneren enthalten diese Klassen die bekannte Programmierung per Structured Text (ST), Anweisungsliste (AWL), Kontaktplan (KOP) nach IEC 61131-3 oder in ANSI-C.

Mit Parametern und Schnittstellen versehen, bezeichnet man diese Programmbausteine als Objekte. Sie lassen sich beliebig zu ganzen Programmen zusammenstellen. Der Vorteil dieser Entwicklungsmethode ist ihre durchgängige Modularität von der untersten Ebene der einzelnen Funktion bis hinauf zum Gesamtprojekt, die durch hierarchische Gruppierung und Verbindung von Programmobjekten wie in einem Baukastensystem entsteht. Das sorgt für Übersichtlichkeit und zwingt zu strukturierter Software-Entwicklung. Der gefürchtete „Spaghetti-Code“ kann gar nicht entstehen.

Vom Konzept bis zum Detail

„Ein wesentlicher Vorteil der objektorientierten Programmierung ist, dass sie zu wohlüberlegter Formulierung der Definitionen zwingt“, findet Franz Aschl. „So erfolgt zuerst die Festlegung der funktionalen Archi-

tektur. Erst wenn diese das OK erhalten hat, wird Arbeit in die Detailprogrammierung gesteckt.“ Dass diese einen geringeren Aufwand als bei traditioneller Programmierung darstellt, liegt nicht nur an den Möglichkeiten zur Wiederverwendung und Eigenschaftsvererbung der Klassen oder an den umfangreichen Bibliotheken mit vorgefertigten Klassen in der Softwarearchitektur Lasal von Sigmatek.

Das größte Potential zur Einsparung von Programmierzeiten liegt darin, dass durch die frühzeitige Überprüfbarkeit nur einmal programmiert werden muss. „Der Clou ist, dass Softwareentwickler und Maschinenbauer wieder in den gleichen Begriffen denken“, sagt Franz Aschl. Die Integration der Automatisierung ist aber nur total, wenn auch Antriebstechnik, Visualisierung, Sicherheitstechnik und Numerik als Teilfunktionen der Gesamtmaschine mit derselben einheitlichen Methodik zu realisieren sind. So kann etwa die frühzeitige Definition der Benutzeroberfläche nicht nur der Abstimmung der Entwicklungsziele mit dem Kunden dienen, sondern auch der Festlegung nachgelagerter Funktionen, die daher auch keine doppelte Dateneingabe brauchen.

Wissenserhalt und Komfort

„Heutige FH- oder TU-Abgänger sind bereits mit objektorientierten Software-Entwicklungsmethoden aufgewachsen“, stellt Franz Aschl fest. „Sie sind es gewohnt funktionsorientiert zu denken und ihre Entwürfe zu überprüfen, bevor sie zu codieren beginnen.“ Dennoch müssten verschiedene Generationen in begrenzter Zeit Automatisierungslösungen mit nachvollziehbar hoher Softwarequalität schaffen, die sich zudem einfach wiederverwenden lassen.

Dazu brauchen sie eine durchgängige Entwicklungsumgebung, innerhalb derer sie über sämtliche Projektphasen hinweg die Funktionalität ihrer Produkte gestalten können. Gefragt ist ein Engineering Tool mit intuitiver Bedienung, komfortablen Möglichkeiten zur Simulation und Visualisierung aller Prozesse, mit einer vollständigen Testumgebung bis hin zum Debugger und mit zentraler Verwaltung von Projekten und Versionen. „Von Sigmatek gibt es diese Software-Umgebung für alle Projektphasen bereits seit zehn Jahren. Sie heißt Lasal“, bekräftigt Franz Aschl.

Sigmatek

www.vfmz.net/3096840