

Objekten gehört die Zukunft

Die in der IT bewährte objektorientierte Programmierung steht in der Automatisierungstechnik noch ganz am Anfang. Aber: Immer komplexer werdende Maschinen und Anlagen erfordern mehr und mehr bei der Softwareerstellung eine Änderung der Vorgehensweise. Althergebrachte Programmiermethoden können die an sie gestellten Anforderungen nicht mehr erfüllen. Aus diesem Grund führt kein Weg an der Objektorientierung in der Automatisierungstechnik vorbei.

von Alfred Sannebeck, Andreas Leu und Martin Buchwitz

VERGLEICHT MAN die Entwicklung der Programmiersprachen in der IT- und in der Automatisierungswelt, so stellt man fest, dass die IT-Welt wesentlich früher die Notwendigkeit erkannt hat, für die steigenden Anforderungen passende Sprachen zu entwickeln. Die Frage ist: Warum

haben sich die Programmiersprachen in der IT-Welt im Gegensatz zu denen in der Automatisierungstechnik so dramatisch und schnell weiterentwickelt? Einen Teil der Antworten findet man dort, wo die Software erstellt wird, also beim Anwender. Dieser betrachtet das Produkt „Software“ aus verschiedenen Perspektiven:

- Das Management ist bestrebt, Software schnell und effizient herstellen zu lassen.
- Das mittlere Management, zum Beispiel die Entwicklungsleitung, muss seine Entwickler so einsetzen, dass sie die gestellte Aufgabe mit möglichst geringem Aufwand umsetzen können. Programme müssen schnell und möglichst fehlerfrei funktionieren.
- Der Programmierer möchte bei der Programmierung getesteten, sicheren Code wiederverwenden. So muss er sich nicht auf das Programmieren von lästigen, immer wiederkehrenden Routinen konzentrieren.
- Beim Kunden ist neben dem Preis wichtig, dass die erstandene Software möglichst lange ohne Fehler läuft.

Um diesem Anspruch gerecht zu werden, haben die Software-Entwickler in der IT bereits vor Jahrzehnten erkannt, dass man Sprachen entwickeln



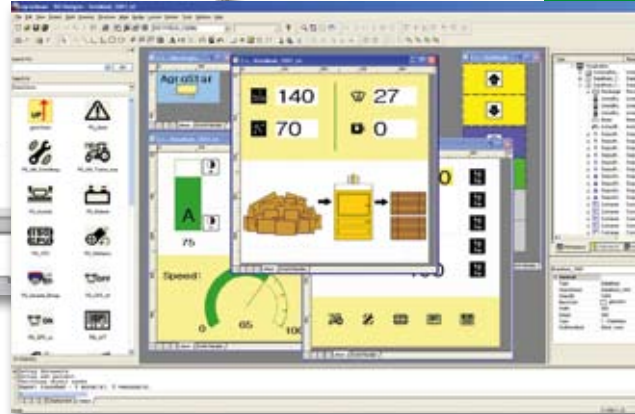
Die zunehmende Komplexität moderner Maschinen erfordert eine effiziente und sichere Programmierung, wie sie die Objektorientierung bietet.

Alle Bilder: Jetter AG

Objektorientierung erobert die Automatisierung



Von der Klasse T3P_Controller wird die Klasse T3P_Controller-Pulse abgeleitet.



Auch bei der Visualisierung ist der objektorientierte Ansatz nicht mehr wegzudenken.

muss, die komplizierte Anweisungskonstrukte in einfachen Befehlen abbilden. Ein großer Meilenstein war in diesem Zusammenhang die vor rund 30 Jahren eingeführte Objektorientierung. Mit ihr lässt sich Programmcode vereinfachen und wiederverwenden. Natürlich kann man hier einwenden, dass Programmierer schon immer Code wiederverwendet haben. Dies geschah jedoch meist durch Copy & Paste oder durch das Wiederverwenden von Funktionen und Unterprogrammen. Die Erfahrung zeigt jedoch, dass gerade hier eine große Fehlerquelle liegt: der kopierte Code wird nicht richtig angepasst oder zwischendurch von Kollegen modifiziert.

In der Automatisierungstechnik lag – im Gegensatz zur IT-Welt – der Fokus ursprünglich stark auf der Funktionalität und Effizienz der Maschine. Über die Effizienz der Programmierung hat sich, wenn die Maschine erst einmal lief, niemand mehr Gedanken gemacht. Im Laufe der Jahre musste ein Automatisierungssystem jedoch neben immer komplexeren, schnelleren Produktionsprozessen auch immer größere Ströme von Produktdaten verarbeiten.

Sicherheit und Effizienz durch Objektorientierung

Mit der Objektorientierung lassen sich bewährte und getestete Routinen so kapseln, dass weder der Code noch die wichtigen Daten verändert werden können. Andererseits kann jeder Pro-

grammierer die Grundfunktionalität erweitern. Die Verwendung dieser Programmteile ist nicht nur effizient, sondern auch sicher.

Der Softwareentwicklung steht damit ein Mittel zur Verfügung, das genau die genannten Ansprüche an eine zeitgemäße Programmiersprache abdeckt. Durch eine Vereinfachung, die einerseits in einfachen Befehlen und andererseits in der sicheren Wiederverwendung getesteter Software-Module besteht, wird die Programmierung nicht nur effizienter, sondern auch deutlich fehlerfreier.

Innovative Programmierung für die Automatisierungstechnik

Die Jetter AG hat von Beginn an eine moderne und einfache Programmiersprache entwickelt. Bereits vor mehr als 25 Jahren konnte der Jetter-Programmierer mit einer Pascal-ähnlichen Sprache den damals komplexen Ablauf einer Achspositionierung durch einen einzigen einfachen Befehl, den Befehl „POSITIONIERE (Position, Geschwindigkeit)“, beschreiben. Die aktuelle Programmiersprache JetSym STX ist eine konsequente Weiterentwicklung davon. Sie basiert

Objektorientierung erobert die Automatisierung

auf dem strukturierten Text und ist damit in der Automatisierungstechnik bekannt. JetSym STX wurde gegenüber der Norm um viele wichtige Elemente erweitert, die in der heutigen Automatisierung zwingend erforderlich sind. Dazu gehört auch die Objektorientierung, die zu einem sehr hohen Grad in JetSym STX integriert ist.

Objektorientierung bei der Visualisierung

Auch auf dem Gebiet der SCADA-Software stellt die Jetter AG mit JetViewSoft eine Software zur Verfügung, bei der der objektorientierte Gedanke konsequent weiterverfolgt wurde. Auch hier ist es möglich, die Effizienz und Sicherheit eines Visualisierungsprogramms durch die Kombination von Objektorientierung und klassischen Visualisierungselementen zu steigern. Anwender von JetViewSoft können zudem mit Hilfe der Programmiersprache JetSym STX als Scriptsprache vielfältige funktionale Erweiterungen ihrer Visualisierung programmieren.

Im Grunde hat jeder, der Automatisierungsanwendungen programmiert, seit Jahren mit Objekten zu tun. Egal, ob es sich um Antriebe, Regler oder um sonstige Aggregate handelt, sie alle werden in Form von Software abgebildet. Und da bietet die objektorientierte Programmierung die passenden Mittel, um diese „Objekte“ in adäquater Weise wiederverwendbar und sicher beschreiben zu können.

Ein objektorientiertes Programm besteht aus strukturierten Objekten. Der Vorteil eines Objekts besteht darin, dass es eine in sich abgeschlossene Einheit von Daten und Algorithmen darstellt. Der Programmierer bestimmt während der Designphase, welche Methoden und Daten von außen sichtbar sind.

Eine Klasse im Sinne der objektorientierten Programmierung ist die Verbindung und Kapselung einer Datenstruktur mit den Funktionen oder Unterprogrammen, die auf dieser Datenstruktur operieren. Sie dient der besseren Wiederverwendbarkeit. Bei der Verwendung im Programm werden dann konkrete Variablen einer Klasse angelegt, so genannte Objektinstanzen.

Im konkreten Vergleich zur bisherigen Welt des Programmierens von Steuerungen lässt sich eine Klasse noch am ehesten mit einem Funktionsblock aus der klassischen SPS-Programmierung (IEC 61131-3) vergleichen. Während ein Funktionsblock jedoch nur eine Funktion besitzt, die den Zustand des Funktionsblocks beeinflusst, kann eine Klasse beliebig viele Funktionen (Methoden) haben. Methoden haben Zugriff auf alle Eigenschaften ihrer Klasse. Es sind also Funktionen, denen ein „unsichtbarer“ Parameter – die aktuelle Instanz – übergeben wird.

Sowohl Objekte als auch Klassen können Eigenschaften und Methoden anderer Objekte beziehungsweise Klassen übernehmen und diese erweitern. So ein Vorgang wird als Vererbung bezeichnet. Gerade diese Fähigkeit ist für die Automatisierungstechnik hochinteressant. Einzelne Teile einer Anlage, zum Beispiel eine Achse, ein Drehtisch oder ein Ventil, kann man nun als Software-Objekt im Steuerungsprogramm eins zu eins abbilden. Typische Anwendungsbeispiele für Objekte sind:

- Regleralgorithmen aller Art
- Logbuchverwaltung
- Beschreibung von Anlagenteilen wie Drehtische, Pumpen, Mischer usw.
- Softwareabstraktion von konkreter Hardware, beispielsweise Achsen
- Abstrahieren der Visualisierung (Datenaustausch und Bedienungsfunktionen)

Mit der Sprache JetSym STX und dem neuen SCADA-Tool JetViewSoft hat die Jetter AG die Möglichkeit geschaffen, Programme mit Hilfe der Objektorientierung auf höchstem Niveau zu realisieren. Dies bringt einen deutlichen Anstieg der Sicherheit und der Effizienz der Programme mit sich. Die Folgen davon sind Einsparungen von Ressourcen und Kosten und eine steigende Kundenzufriedenheit. ■

Alfred Sannebeck ist Produktmanager Software und Steuerungen, Andreas Leu ist für das technische Marketing zuständig und Martin Buchwitz leitet den Bereich Marketing-Kommunikation bei der Jetter AG.