

Modularität ist Trumpf

Wie objektorientierte Programmierung das Engineering erheblich vereinfacht



ENGINEERING-SOFTWARE Bei der Entwicklung moderner Maschinenkonzepte haben die Aufwendungen für Software mittlerweile jene für Mechanik und Elektronik überholt. Gefragt sind daher effiziente Engineering Tools, die Zeit und Kosten sparen. Mit objektorientiertem Software-Engineering hat der Maschinenbauer im Wettbewerb das entscheidende Ass im Ärmel: Modularität.

Sigmathek hat sich bereits vor langer Zeit vom Spaghetticode verabschiedet und setzt seit dem Jahr 2000 konsequent auf die objektorientierte Programmierung (OOP). Die einzelnen Software-Module (Objekte) lassen sich individuell im Baukastensystem zusammenstellen. Wie in der Mechanik, wo eine erprobte Konstruktion wiederverwendet wird, können auch bei der OOP dank der modularen Struktur einmal erstellte Applikationsteile einfach wieder verwendet werden, ohne diese noch einmal überprüfen zu müssen.

Mit Lasal steht dem Anwender ein ausgereiftes und in tausenden Maschinenapplikationen eingesetztes Komplettwerkzeug zur Verfügung, mit dem sich

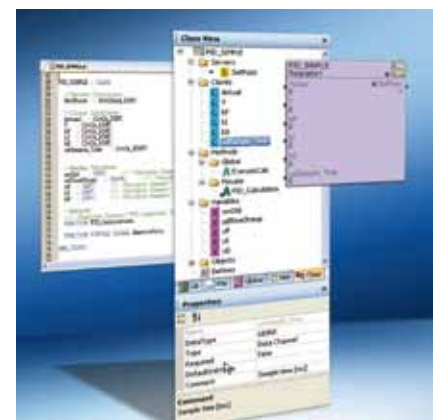
neben der eigentlichen Steuerungsprogrammierung auch Visualisierungs-, Motion Control-, Safety- sowie Service- und Fernwartungs-Aufgaben effizient und komfortabel realisieren lassen.

Nachhaltige Software für komplexe Applikationen

Bei der OOP werden Code und Daten in logische Einheiten („Objekte“) zusammengefasst. Hinter jedem Objekt steht eine „Klasse“. Sie ist der Bauplan des Objekts und definiert den Programmcode sowie die Datenelemente. Jede Klasse übernimmt eine bestimmte Aufgabe, wie beispielsweise die Ansteuerung eines Antriebs oder die Auswertung

einer Temperatur. Beim objektorientierten Ansatz sind die Daten gekapselt und können von außen nicht verändert werden. Die gekapselten Objekte kommunizieren über Schnittstellen mit der „Außenwelt“.

Dank der klar definierten Schnittstellen lassen sich die gekapselten Softwarebausteine mit Leichtigkeit austau-



Über den Klassentree von Lasal CLASS wird die Klasse mit ihren Eigenschaften, Methoden und Schnittstellen definiert, die dazu notwendige Deklaration und grafische Darstellung erledigt das komfortable Engineering Tool aus dem Hause Sigmathek automatisch.

schen. Software wird somit „nachhaltig“, das ist gerade bei komplexen Applikationen ein wichtiger Faktor. Getestete Bausteine werden in sogenannten Klassenbibliotheken abgelegt und können dann in unterschiedlichen Projekten oder Systemteilen einfach wiederverwendet werden.

OOP: Anwenderfreundlichkeit steht im Fokus

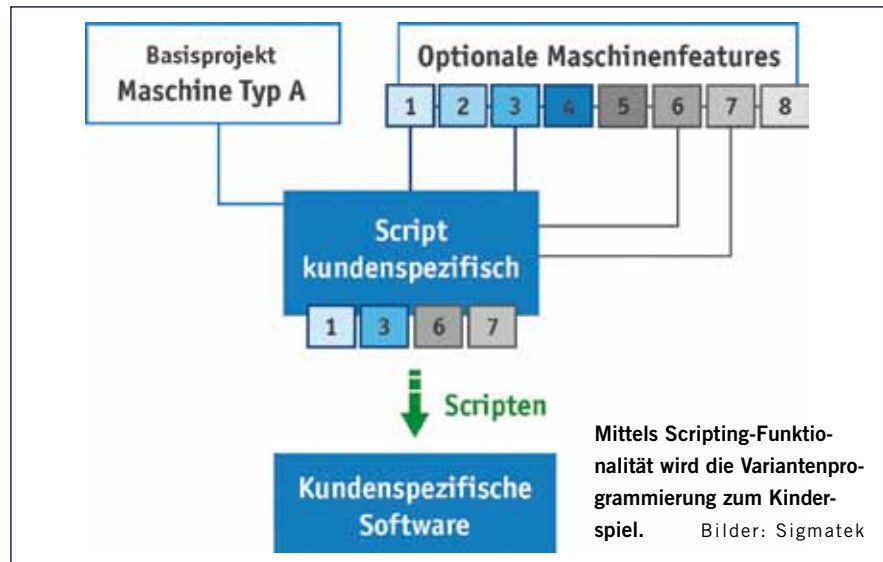
Der Anwender soll alle Vorteile der OOP nutzen können, ohne dass er in der praktischen Anwendung mit der komplexeren Syntax der Objektorientierung in Berührung kommt. Lasal übernimmt alles, was rundum anfällt, wie zum Beispiel die ganze Deklaration der Klassen. Folglich kann sich der Entwickler auf die Implementierung der Methoden (Funktionen) konzentrieren.

Der eigentliche Programmcode des Objektes wird in den gängigen Sprachen der IEC 61131-3 (Strukturierter Text, Kontaktplan, SFC oder C) implementiert. Die Methoden der objektorientierten Programmierung stehen somit als durchgängige Erweiterung der vertrauten (Hoch) Sprachen zur Verfügung.

Für häufig benötigte Funktionen stehen in den Lasal-Bibliotheken vorgefertigte und getestete Funktionsbausteine und Technologiemodule bereit. Beispiele dafür sind Betriebsdatenerfassung oder im Motionbereich interpolierende Bewegungen mehrerer Achsen.

Der Anwender wählt das passende Modul, übernimmt es in sein Projekt und kann, nach dem Setzen einiger Parameter, direkt starten bzw. eine Simulation ablaufen lassen - ohne etwas programmieren zu müssen.

Die grafische Darstellung von Lasal trägt ebenfalls zur Vereinfachung bei. Dabei werden die von Klassen erzeugten



Objekte in sogenannten Netzwerken dargestellt. Der Code selbst ist auf den ersten Blick nicht ersichtlich. Dargestellt werden die Beziehungen der Programmteile zueinander sowie deren wichtigste Daten. Somit können sich Entwickler schnell einen Überblick über die Projektstruktur verschaffen.

Grafische Darstellung garantiert den Überblick

Selbst komplexe Applikationen lassen sich mit dem grafischen Ansatz transparent und übersichtlich darstellen. Zudem ist es Servicetechnikern so rasch und einfach möglich, eine Diagnose über die Fehlfunktion einer Maschine zu treffen.

All diese Faktoren tragen dazu dabei, die Entwicklungszeiten zu reduzieren und sowohl die Entwicklungs- als auch die Gesamtkosten der Maschine über den kompletten Lebenszyklus zu verringern. Davon profitiert nicht nur der Maschinenbauer selbst sondern auch seine Kunden.

Das Lasal Runtime System (kurz LARS) ermöglichte eine windows-basierte Simulation des Codes ohne angeschlossene Steuerung. Dies gilt für komplette Applikationsprogramme genauso wie für noch nicht fertiggestellte Systemteile. So ist es möglich, die Funktionalität der Maschine zu überprüfen und den Quellcode zu debuggen, noch bevor die mechanischen Komponenten fertig sind. Eventuelle Designfehler lassen sich so frühzeitig ausmachen.

Verkürzte Engineeringzeiten und höhere Qualität

Der Anwender wird mit effizienten Diagnose- und Debug-Tools wie Online-Debugger, Echtzeit Data Analyzer, Realtime-Trendaufzeichnungen, Online Diagnose und umfassendem Projektvergleich unterstützt, die auch zu einer Verkürzung der Inbetriebnahme- und Servicezeiten beitragen. Service- und Fernwartungstools wie der Lasal Remote Manager ermöglichen eine globale Wartung und sorgen für einen unterbrechungsfreien Produktionsprozess.

Basierend auf den gekapselten Software-Bausteinen lässt sich durch Einsatz der standardisierten Script-Sprache Python die Software sogar automatisch erstellen oder abändern. So können aus einem Basisprojekt verschiedene Maschinentypen modelliert werden. Ein Beispiel im Infokasten zeigt wie einfach das funktioniert. [in]

Sigmathek
Tel. +43-6274-4321-0

Programme auf Knopfdruck – ein Beispiel

Ein Kunde bestellt den Basistyp A einer Maschine. Bei diesem Typ A hätte er gerne die Option 1, 3, 6 und 7. Die Applikationssoftware für diese spezifische Maschine lässt sich nun vollautomatisch erstellen, ohne dass der Softwareingenieur auch nur eine Programmzeile manuell ändern muss.

Dazu kann er auf eine Bibliothek mit vorgefertigten Klassen zugreifen, die die verschiedenen Module enthalten. Basierend auf dem Grundprogramm werden die für den Typ A spezifischen Module ausgewählt, die gewünschten Sonderfunktionalitäten der Optionen hinzugefügt, und dann wird - gewissermaßen auf Knopfdruck - das Systemprogramm erzeugt, das perfekt mit der Hardware interagiert.