



Bild 1: Das durchgängige Engineering Tool Lasal von Sigmatek unterstützt die modulare Sicht auf die Maschine. Die verschiedenen Maschinenteile und deren Interaktionen spiegeln sich in der Software wider.

Engineering: In Komponenten denken

Der objektorientierte Ansatz erweitert die prozedurale Programmierung, die immer öfter an ihre Grenzen stößt. Was auf den ersten Blick komplex wirken mag, erweist sich mit modernen Engineering Tools wie Lasal von Sigmatek als einfach, transparent und effizient. Der Entwickler erhält so ein hohes Maß an Flexibilität beim Softwaredesign. Ein weiteres Plus: In objektorientierten Konzepten wird genau so programmiert, wie im Maschinenbau gedacht wird – in Komponenten.

Flexible Maschinen- und Anlagenkonzepte erhöhen die Wettbewerbsfähigkeit. Voraussetzung dafür ist, dass die nötigen Produktionsschritte in Module bzw. mechanische und elektrische Komponenten aufgeteilt werden. Diese Module lassen sich dann – ganz nach den spezifischen Kundenanforderungen – individuell zusammenstellen. Und genau dieser Ansatz wird in der objektorientierten Programmierung aufgegriffen. Maschinenteile werden durch Softwarekomponenten (Objekte) nachgebildet. Sigmatek setzt bereits seit dem Jahr 2000 auf Objektorientierung, auch bei der Visualisierung. Mit Lasal steht dem Maschinenbauer ein ausgereiftes und in sehr vielen Maschinenapplikationen erfolgreich eingesetztes Komplettwerkzeug zur Verfügung, mit dem sich neben der eigentlichen Steuer-

ungsprogrammierung auch Visualisierungs-, Motion Control-, Safety- sowie Service- und Fernwartungs-Aufgaben effizient und komfortabel realisieren lassen. Diese methodische Durchgängigkeit in allen Phasen der Softwareentwicklung hilft dem Anwender viel Kosten, Zeit und natürlich auch Nerven zu sparen. Lasal verfolgt seit jeher einen objektorientierten Ansatz. Die Idee dabei ist es, Code und Daten in logische Einheiten, in 'Objekte' zusammenzufassen. Hinter einem Objekt steht jeweils eine so genannte Klasse, die der 'Bauplan' für ein Objekt ist und daher den Programmcode und die Datenelemente definiert. Jede Klasse übernimmt eine bestimmte Aufgabe, wie beispielsweise die Ansteuerung eines Antriebs, eines Ventils oder die Ausführung eines PID-Regelalgorithmus. Der eigentliche Programmcode

eines Objektes wird in den gebräuchlichen Sprachen der IEC 61131-3, z.B. Strukturiertem Text, Anweisungsliste oder Kontaktplan implementiert. So stehen die Methoden der objektorientierten Programmierung als durchgängige Erweiterung der vertrauten und bewährten Sprachen zur Verfügung.

Softwaremodule einfach wiederverwenden

Die einzelnen Software-Module (Objekte) lassen sich wie in einem Baukastensystem flexibel zusammensetzen. Dank dem modularen Aufbau lassen sich – wie in der Mechanik, wo auf erprobte Konstruktionen laufend zurückgegriffen wird, – auch bei der OOP einmal erstellte Applikationsteile einfach wiederverwenden, ohne diese noch einmal überprüfen zu müssen.

Bei vielen Projekten existieren ähnliche Funktionalitäten, natürlich in unterschiedlichen Ausprägungen wie beispielsweise die Ansteuerung von verschiedenen Antriebsarten. Durch Vererbung entsteht eine neue Klasse, welche eine zugewiesene Basisklasse verfeinert (d.h. spezialisiert), indem sie mit zusätzlichen Informationen und Programmcodes erweitert und angepasst wird – um beim Beispiel zu bleiben 'Servodrive oder Schrittmotor'. Einzelne Klassen lassen sich zu einer komplexen Klasse zusammenfassen – der Fachausdruck dafür heißt 'Aggregieren'. Mit diesen Techniken der OOP ist es möglich, neue Ausprägungen von Maschinenteilen mit minimalem Programmieraufwand umzusetzen. Wird ein objektorientierter Code verändert, ist es aufgrund der definierten Schnittstellen in den meisten Fällen ausreichend, nur die geänderten Klassen zu testen anstatt große Teile des Projekts. All dies reduziert die Engineeringzeiten und -kosten erheblich, zudem wird das Fehlerisiko reduziert.

Strukturiertes Maschinendesign

Ein großes Plus der OOP ist, dass hier genauso wie im Maschinenbau in Komponenten gedacht wird, also beispielsweise in Motoren, Getrieben und daraus folgenden Antriebssträngen. Jeder Applikationstechniker, der eine Maschinensoftware entwickelt, muss sich im Vorfeld detailliert mit der Maschine auseinandersetzen. Welche Teile gibt es auf der Maschine, wie hängen die verschiedenen Maschinenteile zusammen, wie beeinflussen sie sich gegenseitig, und wie interagieren sie miteinander. Durch diese Analyse der Maschine und die Definition der einzelnen Maschinenteile ergeben sich bereits die Klassen samt ihren Eigenschaften und den Schnitt-

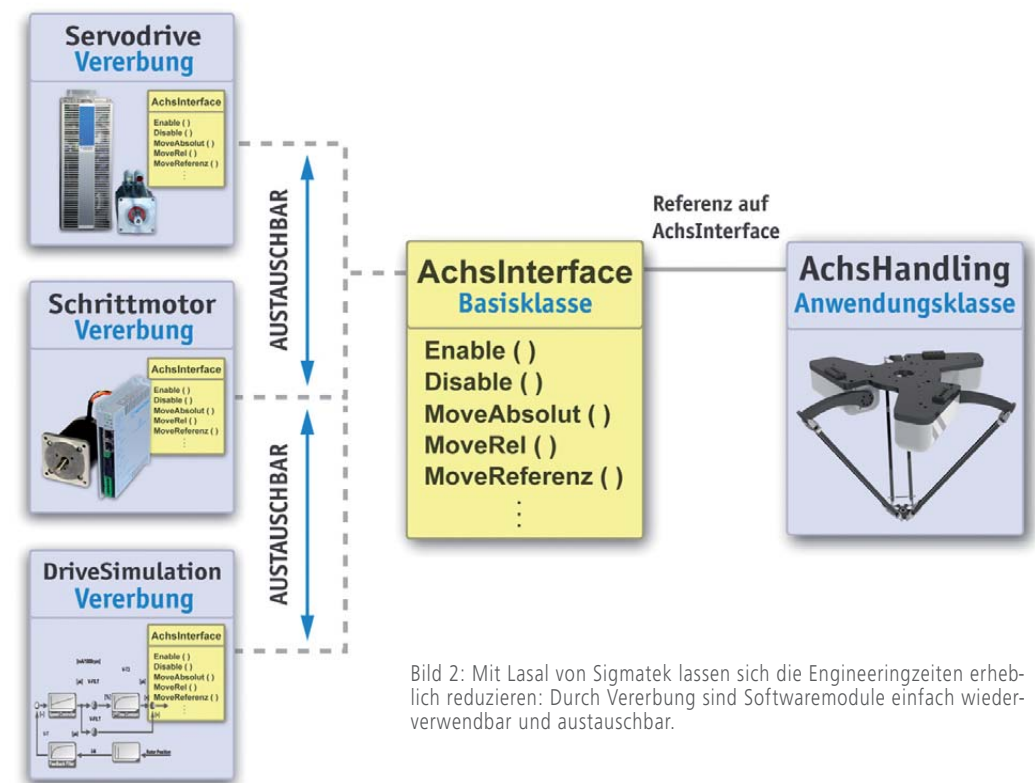


Bild 2: Mit Lasal von Sigmatek lassen sich die Engineeringzeiten erheblich reduzieren: Durch Vererbung sind Softwaremodule einfach wiederverwendbar und austauschbar.

stellen zu den anderen Maschinenteilen. So werden auch Maschinenteile, die sich gleich oder ähnlich verhalten, analytisch extrahiert und können in allgemeinen Klassen beschrieben werden. So erfolgt das Softwaredesign strukturierter und ist gleichzeitig einfacher, da nicht in komplizierten Codes sondern in Objekten bzw. Komponenten gedacht wird. Ein wichtiges Feature der objektorientierten Programmierung ist die Vererbung. Nehmen wir als Beispiel die Ansteuerung von Antrieben. Je nach Maschinenvariante ist es erforderlich, unterschiedliche Antriebsarten anzusteuern, wie etwa einen Servomotor, einen Schrittmotor oder eine reine Simulation der Verfahrbewegung für Präsentationszwecke. Die Klasse 'Achs-Interface', implementiert die Bewegungs-

abläufe des Antriebs je nach Maschinenzustand. Diese Klasse setzt Kommandos an den Antrieb ab wie 'Enable()', 'FahreReferenz()' oder 'MoveToPosition()'. Softwaretechnisch am wenigsten zielführend wäre es, die verschiedenen Antriebsarten schon in der Klasse für den Bewegungsablauf zu berücksichtigen, da die Software so komplexer und schlechter test-beziehungsweise wartbar wäre. Die Objektorientierung ermöglicht hier mittels Vererbung eine einfache Abhilfe: Es wird eine Schnittstellenklasse definiert, in der alle Kommandos eingefügt werden, die die Klasse „Achs-Handling“ benötigt. Diese Schnittstellenklasse „Achs-Interface“ kann als Basisklasse für die un-

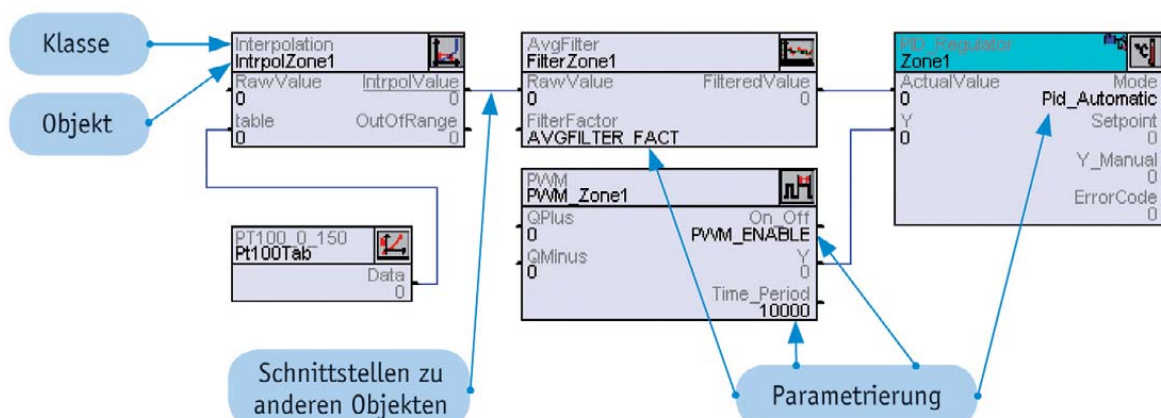
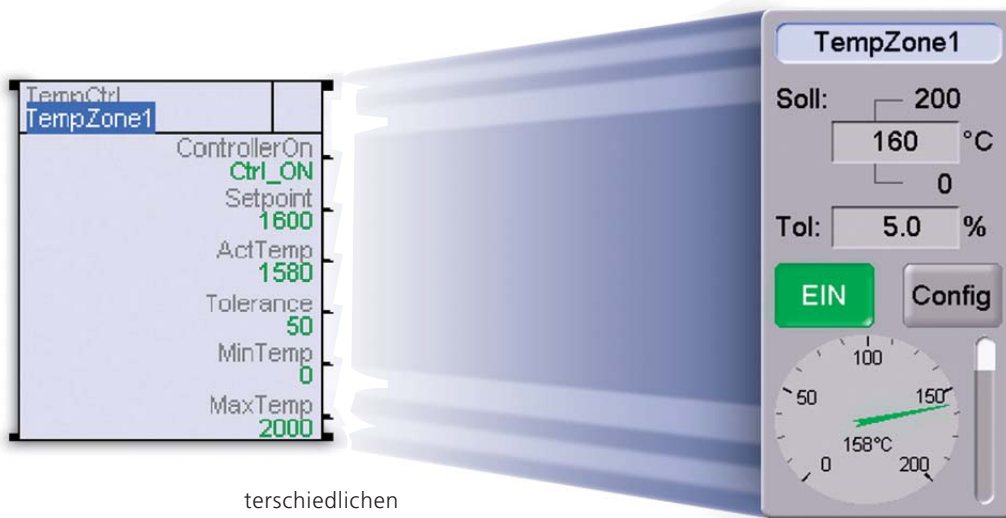


Bild 3: Ein Rechteck auf diesem Netzwerk stellt ein Objekt dar, oben der Klassenname und darunter der Name des daraus erzeugten Objekts. Die Objekte haben Schnittstellen zur Außenwelt, um mit anderen Objekten zu kommunizieren. Parametrierung: Diese Schnittstellen sind zugleich auch Datenelemente, die für jedes Objekt unterschiedlich initialisiert werden können.



verschiedlichen Antriebsvarianten dienen. Die Implementierung der einzelnen Methoden wird dann in der Ableitung (vererbte Klasse) ausprogrammiert, da die Enable()-Funktion für einen Servoantrieb klarerweise anders aussieht, als für die Simulation. Das Wesentliche an diesem Beispiel jedoch ist, dass die Achs-Handling-Klasse immer nur auf die Basisklasse referenziert. Somit wird der tatsächlich angeschlossene Antrieb softwaretechnisch einfach austauschbar. Dies ist ein recht einfaches Beispiel, das aber für jede Maschinenfunktion Gültigkeit hat. Die Objektorientierung hilft bei der Strukturierung und beim Design der Maschinensoftware, da reale Teile in Klassen abgebildet werden. So wird das Engineering beschleunigt und stark vereinfacht. Die vom Programmierer definierten Bausteine werden in sogenannten Klassenbibliotheken (Libraries) abgelegt und können dann in unterschiedlichen Systemteilen oder anderen Projekten wiederverwendet werden. Dazu steht bei Lasal eine Vielzahl an Funktionsklassen für unterschiedliche Anwendungskategorien bereits in Standardbibliotheken zur Verfügung.

Übersichtliche Maschinensoftware

Bei der OOP kommt dem Software-Design eine große Bedeutung zu, d.h. es hat mindestens den gleichen Stellenwert wie die Implementierung. Der Fokus liegt dabei auf den abgeschotteten Objekten, die über Schnittstellen mit der Außenwelt kommunizieren. Wenn sich der Anwender diese Schnittstellen vorab genau überlegt, kann er Objekte später leicht gegen andere austauschen. Die Maschinensoftware wird damit sehr flexibel, wartungsfreundlich und leicht testbar – und dies

noch nach Jahren. Vor allem bei komplexen Projekten kommt diese Strukturiertheit der OOP dem Maschinenbauer zu Gute. Je umfangreicher das Projekt ist und je mehr Personen über den Produktlebenszyklus eingebunden sind, desto größer ist der Vorteil, der sich durch die Übersichtlichkeit der Software dank klarer Kapselung ergibt. Diese Übersichtlichkeit wird durch die grafische Darstellung noch verstärkt. In Lasal Class, der Entwicklungsumgebung von Sigmatek wird der grafische Ansatz von jeher verfolgt. Dabei werden die von Klassen erzeugten Objekte in sogenannten Netzwerken dargestellt. Der Vorteil der grafischen Darstellung ist, dass die Maschine in der Software grafisch nachgebildet wird. So sieht man auf einen Blick die Eigenschaften eines Maschinenteils sowie die Kommunikation mit anderen Objekten d.h. Maschinenteilen. Eine echte Stärke der grafischen Darstellung in Lasal zeigt sich auch bei der Online-Diagnose. Sobald eine Onlineverbindung zur SPS besteht, werden im Netzwerk die tatsächlichen Werte der Schnittstellenvariablen angezeigt.

Objektorientierung auch beim Visualisieren

Die Objektorientierung eröffnet nicht nur im Bereich der Ablaufprogrammierung von Maschinen sondern auch im Visualisierungsbereich neue Möglichkeiten. Mit der Visualisierungsumgebung Lasal Screen lassen sich grafische Objekte definieren. Dabei verwendet der Anwender Datenpunkte einer Klasse aus Lasal Class, um deren visuelles Erscheinungsbild zu definieren.

Moderne Tools für Motion Control und Safety

Das all-in-one Engineering Tool bietet mit dem voll integrierten Werkzeug 'Lasal Motion' auch einen umfassenden Funktionsbaukasten zur Steuerung von Antrieben. Viele Positionierungsmöglichkeiten, CNC-Funktionen, koordinierten Bewegungen und Technologiefunktionen sind darin enthalten, die von Kurvenscheiben bis hin zu Bahnsteuerungen mit Transformationen für verschiedene Roboter-Kinematiken reichen. Die Integration von SPS, Motion Control und Visualisierung führt zu einer verbesserten Synchronisation von Prozess- und Bewegungsabläufen in der Maschine. Daraus ergeben sich eine höhere Produktionsgeschwindigkeit, Präzision und Produktqualität. Für die Programmierung und Konfiguration der Safety-Komponenten steht ebenfalls das passende Werkzeug zur Verfügung: der nahtlos integrierte Lasal Safety Designer. Auf Basis einer Funktionsbibliothek, die neben Standard-Funktionsblöcken auch Safety-Funktionsblöcke angelehnt an den PLCopen-Standard enthält, kann der Anwender I/O-Konfigurationen und die logischen Verknüpfungen der sicherheitsbezogenen Abläufe einfach und komfortabel erstellen.

Service inklusive

Das Paket Lasal Service rundet die umfassende Softwaresuite von Sigmatek ab. Hier stehen Tools zur Fernwartung und Diagnose wie beispielsweise der Lasal Remote Manager bereit. Mit Hilfe von Servertechnologie kann damit von jedem geografisch entfernten Rechner auf die angeschlossene Steuerung zugegriffen werden. Ein OPC-Server steht als Open Data-Interface zur Verfügung. So ist der individuelle Datenaustausch mit externen Programmen über standardisierte Schnittstellen möglich. ■

www.sigmatek-automation.com



Autor: Bernhard Gangl,
Abteilungsleiter Software,
Sigmatek GmbH & Co KG