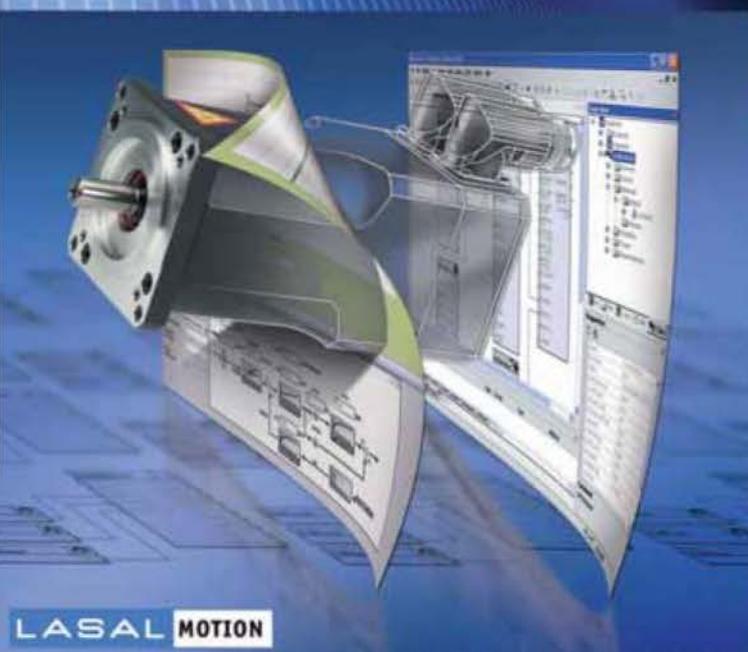


# Um Klassen effizienter entwickeln

In Zeiten kürzer werdender Entwicklungszeiten ist Effizienz nicht nur in der fertigen Automatisierungslösung gefragt, sondern ein entscheidender Wettbewerbsvorteil bereits in der Softwareentwicklung. Diese – und dazu die Möglichkeit, durch modularen Aufbau der Automatisierungssoftware einmal entwickelte Lösungen mit wenig Aufwand kundenindividuell anzupassen, wird mit objektorientierter statt sequenzieller Programmierung erreicht. Mit LASAL bietet Sigmatek eine Entwicklungsumgebung, die auf Basis von Softwareklassen von der Projektierung bis zur Detailprogrammierung komfortabel und effizient alle Ebenen des Entwicklungsprozesses einschließt.

*Autor: Ing. Peter Kemptner / x-technik*

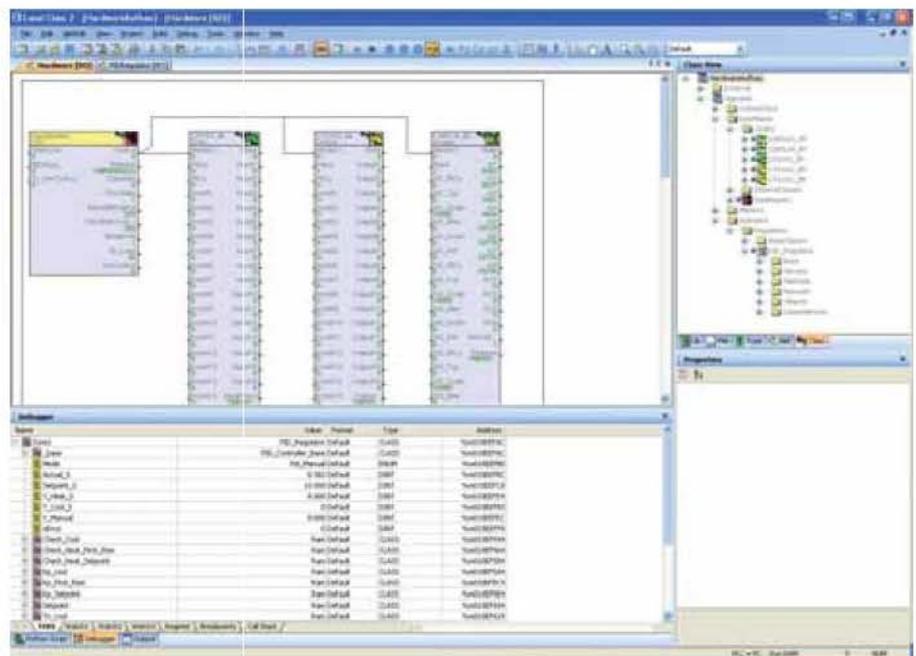


Entwickler von Maschinen und Anlagen – und damit von Automatisierungslösungen – stehen unter enormen Innovationsdruck. Immer kürzer werdende Entscheidungszeiträume für Großinvestitionen machen die Time to Market, also das Erreichen der Entwicklungsziele in kurzer Zeit, zum entscheidenden Kriterium für Erfolg oder Misserfolg. Zudem werden auch Serienmaschinen zunehmend individueller, sodass man in der Branche von „Stückzahl 1“ spricht.

Um diesen Anforderungen zu genügen, geht die Entwicklung zwei Wege: Zum einen werden Maschinen und Anlagen modular aufgebaut, um wie bei einem Baukastensystem durch Weglassen oder Austausch von Komponenten rasch ein den jeweiligen Anforderungen optimal angepasstes Ergebnis zu erzielen, zum anderen wird eine möglichst große Durchgängigkeit der Lösung ohne Systembruch von der Projektierung der Gesamtanlage bis hinunter in die kleinste Komponente angestrebt. Nur so können Kompatibilitätsprobleme der einzelnen Baugruppen untereinander vermieden und das Ergebnis als Gesamtwerk bereits während der Entwicklung überprüft werden. Das gilt für die mechanische Entwicklung aber auch und in besonderem Maß für die Entwicklung der Steuerungs- und Automatisierungssoftware.

### Modularität als Prinzip

Während in anderen Branchen der Softwareentwicklung bereits vor etwa 15 Jahren eine Abkehr von strikt se-



Objektorientierte Softwareentwicklung erfolgt durch Anordnung und Verbindung von Funktionsblöcken. Hier ein typischer Steuerungsaufbau mit CPU, digitalen Ein- und Ausgängen und einem analogen Mischer. Bild: Sigmatek

quenzieller Programmierung erfolgte, prägte im naturgemäß konservativeren Maschinen- und Anlagenbau noch bis vor kurzem die klassische SPS-Programmierung das Denken vieler Steuerungsentwickler. Das ändert sich, seit objektorientierte Programmierung auch in die Steuerungstechnik Einzug hält.

Als erster Anbieter in diesem Bereich hat Sigmatek bereits vor einigen Jahren mit der Projektierungssoftware LASAL die objektorientierte Programmierung mit grafischer Darstellung der Codeblöcke und modularem Aufbau in die Steuerungstechnik übertragen.

Nun stellte der Automatisierungssystemhersteller aus Lamprechtshausen bei Salzburg die zweite Version von LASAL CLASS vor, die mit noch größerer Übersichtlichkeit und Durchgängigkeit zur weiteren Effizienzsteigerung in der Softwareentwicklung beiträgt und zudem die Verarbeitungsgeschwindigkeit maximiert.

Grundprinzip des Projektierungs- und Softwareentwicklungstools ist eine durchgängige Modularität der einzelnen Programmblöcke von der untersten Ebene der einzelnen Funktion bis hinauf zum Gesamtprojekt. Dieses ent-

↳ Fortsetzung Seite 14



LASAL deckt als Software-Entwicklungsplattform alle Bereiche des Produktlebenszyklus ab, von der Top-Down Systementwicklung bis zur maschinen-nahen Programmierung, Parametrierung und Wartung.  
Bild: Sigmatek

steht durch hierarchische Gruppierung und Verbindung von Programmobjekten wie in einem Baukastensystem. Das sorgt für Übersichtlichkeit und zwingt zu strukturierter Software-Entwicklung. Der gefürchtete „Spaghetti-Code“ kann gar nicht entstehen, dennoch ist eine hohe Performance im fertigen System gewährleistet.

#### Objekte, Klassen, Bibliotheken, Clients und Server

Zunächst ein paar Begriffsbestimmungen für diejenigen, die mit der Sprache der objektorientierten Programmierung noch nicht so vertraut sind:

Klassen sind fertige, in sich abgeschlossene (gekapselte) Programmbausteine oder Vorlagen, deren innerer Programmcode eine bestimmte Funk-

tionalität repräsentiert. Zwecks besserer Übersichtlichkeit und Wiederverwendbarkeit in unterschiedlichen Projekten oder Systemteilen werden Klassen in sogenannten Klassenbibliotheken verwaltet. Selbstverständlich sind in LASAL bereits vom Hersteller aus umfangreiche Bibliotheken mit gängigen Funktionsklassen enthalten.

Beim Aufbau eigener Klassen herrscht eine große Freizügigkeit bezüglich der Programmierung. Structured Text (ST) kann ebenso verwendet werden wie Anweisungsliste (AWL), Kontaktplan (KOP) nach IEC 61131-3 oder direkte Programmierung in ANSI-C.

Objekte sind Bausteine, die vom Softwareentwickler durch individuelle Parametrierung und durch Verbindung von und zu anderen Objekten mit ei-

nem eigenständigen Verhalten versehen wurden. Objekte entstehen aber nicht nur durch Parametrierung und Verbindung von Klassen, sondern auch durch Zusammenfassung mehrerer Objekte und/oder Klassen zu größeren Blöcken. Dabei können die Klasseneigenschaften jedes Objekts „vererbt“ werden. So kann jede hierarchische Programmebene die Eigenschaften der untersten Ebene nutzen, aber auch verändern.

Sind komplexe Objekte oder Klassen mit ihren enthaltenen Klassen fertig gestellt und in sich getestet, genügt es, nur die Gesamtobjekte mit den relevanten Schnittstellen nach außen zu betrachten. Auf diese Weise können hoch komplexe Programmstrukturen modelliert und programmiert werden, ohne in das Innere der bewährten



Software-Leiter Ing. Bernhard Gangl: „Von der Projektierung über die Funktionsgruppenentwicklung bis zum Ausprogrammieren einzelner Funktionen sieht der Entwickler mit LASAL CLASS nur die Ebene, die für ihn relevant und beherrschbar ist. Gemeinsam mit der grafischen Programmierung erhöht schon das Effizienz und Sicherheit erheblich.“  
Bild: Sigmatek



LASAL unterstützt Softwareentwickler mit zahlreichen Analysetools. Bild: Sigmatek

Komponenten eingreifen zu müssen. Das ist besonders dann hilfreich, wenn Techniker zur Inbetriebnahme und Einstellung einer Anlage verwendete Objekte parametrieren und verknüpfen, nicht aber deren inneres Verhalten verändern sollen.

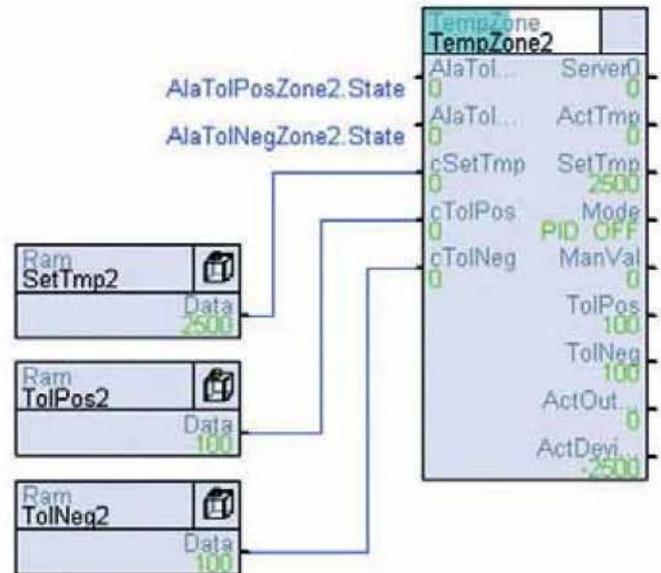
Die Kommunikation der einzelnen Softwarebausteine untereinander erfolgt nach dem Prinzip von Client und Server. Dabei fordert der Client als aktiver Teil der Kommunikation vom Server Information an, die dieser passiv zur Verfügung stellt. Diese Methode ist dafür verantwortlich, dass im Vergleich zu konventionellen Systemen CPU-Leistung eingespart werden kann. Das deshalb, weil Bausteine event-getriggert arbeiten können, also nur dann, wenn sie durch ein äußeres Ereignis wie das Schreiben eines Sollwerts durch die Visualisierung angestoßen werden, während sie dazwischen „schlafen“ und keine Leistung verbrauchen.

### Top-Down Entwicklung

Die eigentliche Mächtigkeit von LASAL CLASS liegt in der Möglichkeit, die Projektentwicklung „von oben nach unten“ zu betreiben. Damit ist der Projektleiter in der Lage, die gesamte Automatisierungslösung zu projektieren und damit wesentliche Teile der Systementwicklung im selben Zug durchzuführen, auch wenn noch nicht alle später benötigten Programmteile existieren. Zum Vorteil, dass nicht in der Softwareentwicklung die Projektierungsabsicht später zeitaufwändig nachvollzogen werden muss, kommt die frühzeitige Überprüfbarkeit der Gesamtfunktionalität.

Praktisch geschieht das durch Erstellung einer Baumstruktur, in die vorhandene oder vorerst leere, aber definierte Klassen eingehängt werden. Die von LASAL selbstständig aus der Struktur abgeleitete Klassendefinition ist die Vorgabe für die einzelnen Entwickler, die anschließend die leeren Blöcke mit Funktion füllen. Hilfreich ist dabei auch die Reichhaltigkeit der mitgelieferten Bibliotheken, die Klassen für Betriebssystemfunktionen, Hardwarezugriffe und Standardklassen wie PID-Regler oder Data Logger gebrauchsfertig zur Verfügung stellen. Noch nicht fertiggestellte Systemteile können mit LARS, einem windows-basierten Simulationstool, simuliert werden. So muss mit Tests fertiggestellter Softwarekomponenten nicht auf das Vorhandensein sämtlicher Teile gewartet werden, was die Entwicklungszeit weiter verkürzt.

Aus fertigen Bausteinen können mit LASAL CLASS vollautomatisch Projekte erstellt oder Anwendungen parametrieren werden. Dazu bedient sich die Software der standardisierten Scriptsprache Python. Mittels verhältnismäßig einfacher Scripts können so auf Knopfdruck abgestimmte Softwareversionen für unterschiedliche Ausstattungsvarianten oder Ausprägungen für kundenspezifisch angepasste Serienmaschinen entstehen, ohne dafür einerseits die Softwareentwicklung selbst zu bemühen oder andererseits ein Risiko bei der Funktion eingehen zu müssen. Wichtig ist in jeder Softwareentwicklung, auch bei objektorientierter Programmierung, die Möglichkeit, die fertigen Programme zu analysieren, Fehler zu suchen und zu beseitigen. Für eine zügige Entwicklung muss dieser Vorgang auf allen Ebenen der Softwareerstellung wiederholt werden, von der einfachen



*Klassen und Objekte sind in beliebiger Tiefe hierarchisch aufgebaut und vererben ihre Eigenschaften auf die nächste Ebene. Objekte in einem eventgetriggerten Client-Server-System sind nur aktiv, wenn sie durch äußere Ereignisse angestoßen werden und sparen dadurch CPU-Kapazität. Bild: Sigmatek*

Funktion bis zum komplexen, parallelisierten Zusammenspiel großer Blöcke im Gesamtsystem. LASAL CLASS stellt dem Entwickler zu diesem Zweck eine Reihe von Tools zur Verfügung, vom Online-Debugger über PlcTraceview Aufzeichnungsmöglichkeiten für das Taskverhalten. Auch ein Oszilloskop ist ebenso Teil des Entwicklungspaketes wie die Analyse des Zeitverhaltens im Multitasking.

### Durchgängigkeit bringt Einsparungspotenzial

Die Entwicklungsumgebung LASAL deckt mit ihren Einzelpaketten alle Aspekte der Automatisierung ab. Neben LASAL CLASS steht mit dem LASAL SCREEN Editor ein Werkzeug für die effiziente Erstellung der Visualisierung zur Verfügung. Das im letzten Heft vorgestellte LASAL MOTION vereinfacht mit seinem modularen Aufbau die Softwareentwicklung rund um die Antriebstechnik. Fernwartung, Software-Updates und Datenaustausch werden mit den LASAL SERVICE Tools (Remote Manager, OPC-Server, Webserver etc.) realisiert.

Einsparungspotenziale ergeben sich aus der Einsatzfähigkeit von LASAL auf allen Zielplattformen der Sigmatek-Produktpalette, die auch einen Wechsel der Plattform ohne Softwareänderung zulässt. Der wirkliche Vorteil liegt aber in der Verwendung einer einzigen Entwicklungsumgebung für sämtliche Projektphasen, im Prinzip von der Pflichtenhefterstellung über die Detailprogrammierung, über die Inbetriebnahme hinaus bis zur Softwarewartung. Für alle Projektphasen eine Software mit Klasse.

### 📞 KONTAKT

**SIGMATEK GmbH & Co KG**  
Sigmatekstraße 1  
A-5112 Lamprechtshausen  
Tel. +43-6274-4321-0  
[www.sigmatek.at](http://www.sigmatek.at)