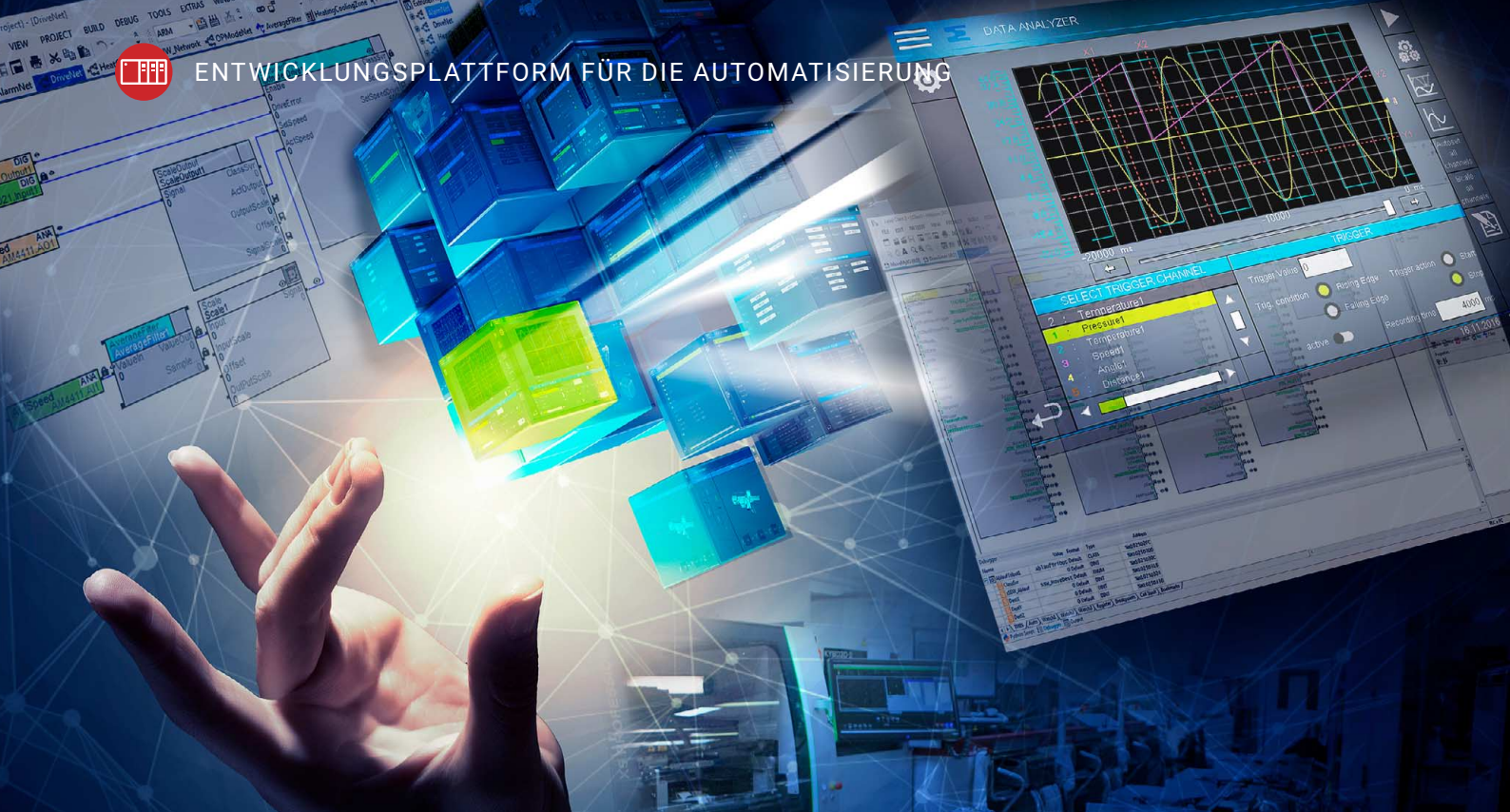




ENTWICKLUNGSPLATTFORM FÜR DIE AUTOMATISIERUNG

*Objektorientierte Programmierung
und grafische Darstellung*

Nicht Zukunft sondern Gegenwart

Mit einer objektorientierten Engineering-Plattform lassen sich Maschinenkonzepte sehr strukturiert, modular und schnell realisieren. Das sind wesentliche Vorteile, speziell wenn es komplexer wird und mit modularer Hardware Schritt gehalten werden soll. Das Engineering Tool Lasal von Sigmatek kommt dabei zudem mit weniger Code aus und sorgt für Übersichtlichkeit mit grafischer Darstellung.

Klar: Für einfache Anwendungen mit wenig Codezeilen braucht es keinen objektorientierten Ansatz, da kommt der Maschinenbauer auch mit konventioneller Programmierung problemlos ans Ziel. Wenn es aber komplexer und smarter wird, ist die objektorientierte Programmierung (OOP) die richtige Wahl, speziell für Serienmaschinen in unterschiedlichen Ausstattungsvarianten und der Anforderung, diese Maschinen in einen intelligenten Maschinenverbund einzubinden. Dieses Programmierkonzept steht für hohe Modularität, Strukturiertheit und einfache Wiederverwendbarkeit der Applikation – so kann softwareseitig mit der Modularität und Skalierbarkeit der Hardware Schritt gehalten werden. Objektorientierte Programmiermethoden sind inzwischen auch im Maschinenbau angekommen. Speziell in der jüngeren Generation der Automatisierungstechniker war OOP zumeist schon Teil der Ausbildung. Es gibt also schon zahlreiche Softwarearchitekten, die Erfahrung mit dieser modernen Programmierart haben und Maschinensoftware objektorientiert in Funktionen und Schnittstellen denken.

Low Code – weniger Aufwand

Bereits seit 20 Jahren setzt Sigmatek auf Objektorientierung. Das Engineering Tool Lasal nach IEC61131-3 ist von Grund auf objektorientiert und zudem als Low-Code-Plattform entwickelt worden, die dem Anwender Programmieraufwand spart. Wenn wie bei Lasal die Objektorientierung mit grafischer Darstellung umgesetzt wird, steigen Übersichtlichkeit und Wartbarkeit des Softwareprojektes. Das ist ein deutlicher Mehrwert für den Maschinenbauer, denn es ist meist die Software, über die er heute den Großteil seiner Wertschöpfung erzielt. Zudem kann die Applikation mit Lasal durchgängig entwickelt werden: Steuerung, Visualisierung, Motion, Safety inklusive Service Tools. OOP eröffnet dem Maschinenbauer neue Möglichkeiten und Wege, erfordert aber eine andere Herangehensweise bei der Applikationsentwicklung. Ausgehend vom Gesamtprojekt überlegen die Softwarearchitekten, welche Funktionen in der Maschine benötigt werden und wie diese miteinander verbunden sind. Beim Top-down-Design geht darum, Aufgaben zu

Bild: Sigmatek GmbH & Co KG



zerlegen und visuell dazustellen: Welche Maschinenteile sprechen mit anderen, welche tauschen Daten aus. Objekte und Schnittstellen werden definiert und im Netzwerk angezeigt. „Es ist keine Zeile Code nötig, um in einem Lasal-Netzwerk, die Funktionsweise der Anwendung zu verstehen“, erklärt Franz Aschl, Innovationsmanager bei Sigmatek.

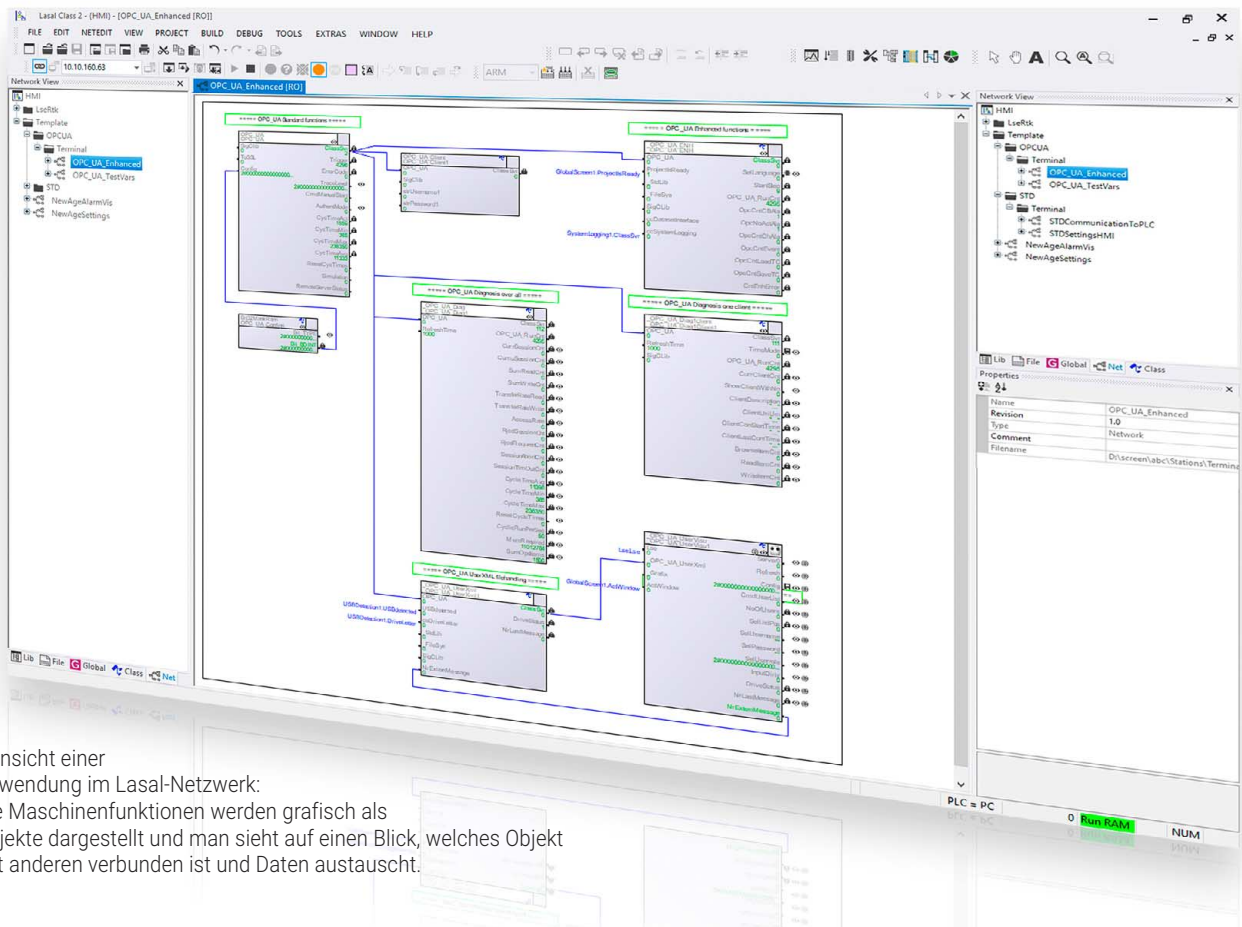
Software im Baukastensystem

Wenn die Funktionen und die Kommunikationsschnittstellen festgelegt sind, beginnt die Entwicklung Bottom-up. Die Grundfunktionen einer Maschine lassen sich mithilfe der umfangreichen Lasal Libraries mit wenigen Klicks zusammenstellen, mit Parametern und Schnittstellen versehen und schon kann der Softwarearchitekt die Anwendung testen. Über 10.000 Klassen umfassen die Lasal-Bibliotheken. Zudem stehen moderne Templates sowie einsatzbereite, getestete Softwarekomponenten bereit: funktionspezifische Add-Ons und themenbezogene Packages. Beispiele sind Regler, Rezeptverwaltung, Alarmsystem, Safety, Datenanalyse aber auch komplexe Motionfunktionen wie Multiachsensteuerung oder Deltaroboter-Inbetriebnahme. „In letzter Zeit ist speziell die für die Kommunikation zuständigen Bibliothek stark gewachsen und reicht von TCP/IP, über OPC UA Client sowie Server und MQTT hin zu SSL-Verschlüsselung“, so Franz Aschl. Es gebe einen vermehrten Bedarf an Vernetzung – horizontal z.B. mit Kamerasystemen oder Steuerungen anderer Hersteller und vertikal mit höheren Systemen wie ERP, MES, Cloud. Mit Einsatz der vorgefertigten Softwarekomponenten lässt sich der Entwicklungsaufwand für Grundfunktionen um bis zu 70 Prozent reduzieren. Der Applikationsingenieur kann sich auf die Implementierung maschinenspezifischer, innovativer Features konzentrieren.

Größtmögliche Transparenz

Lasal ist auch für agile Entwicklungsteams geeignet, da die Arbeit sehr strukturiert erfolgen muss. „Bei unsauber implementierten Programmen und Variablen, die kreuz und quer im Projekt verteilt beschrieben werden, sind Auswirkungen einer Programmänderung praktisch nicht vorhersehbar“, betont Aschl. Ganz anders bei der OOP: Die Objekte werden nach außen hin gekapselt, sodass eine ungewollte Beeinflussung nahezu ausgeschlossen

werden kann. „Variablen sind nur über zugehörige Methoden manipulierbar. Somit sind klare Schnittstellen vorgegeben, die es dann auch zu verwenden gilt“, so Aschl weiter. „Die visuelle Darstellung der gekapselten Objekte im Netzwerk mit allen Kommunikations-Schnittstellen ist bei Lasal einzigartig und sorgt für hohe Transparenz genauso wie das Low-Code-Prinzip.“ „Durch die grafische Darstellung von Objekten wird der Texteditor anders als bei anderen objektorientierten Tools wirklich erst am Schluss eingesetzt“, erklärt Aschl. Die Implementierung der Methoden in der Klasse kann der Softwareingenieur in Programmiersprachen wie ST, AWL oder KOP nach IEC61131-3 oder in C vornehmen. Moderne Engineering Tools wie Lasal machen die Nutzung der objektorientierten Programmierung einfach, indem sie immer wiederkehrende Funktionen übernehmen. Z.B. wird der Code für die Deklarationen von Klassen, Variablen oder Schnittstellen automatisch im



► Ansicht einer Anwendung im Lasal-Netzwerk:
Die Maschinenfunktionen werden grafisch als Objekte dargestellt und man sieht auf einen Blick, welches Objekt mit anderen verbunden ist und Daten austauscht.

Hintergrund erstellt. „Variablen können nicht nur visualisiert, sondern über globale WAN-Netzwerke firmenübergreifend zur Verfügung gestellt werden, sodass auch gleich bei der Entwicklung festgelegt wird, welche Schnittstellen von außen bedienbar bzw. sichtbar sind“, sagt Aschl.

Visuelles Live-Debugging

Das realitätsgetreue Verhalten der Komponenten im Lasal-Projekt ist im Online-Modus in Echtzeit zu sehen. In der Detailansicht sind mögliche Steckplätze und Busverbindungen ersichtlich, aktuelle Werte und Eigenschaften der Komponenten kön-

ger samt zentraler Verwaltung von Projekten und Versionen lässt sich die Transparenz und die Qualität der Software steigern. „Bei Lasal ist sogar ein visuelles Live-Debugging möglich, d.h. die aktuellen Werte sind nicht nur im Programmcode, sondern sofort und live in der grafischen Objektabbildung sichtbar“, führt Aschl weiter aus. „Das spart enorm viel Zeit.“

Evolution statt Revolution

„Wer nicht an Objektorientierung glaubt, ist von gestern“, meint Aschl etwas provokant aber mit einem Augenzwinkern. Objekte entsprechen Maschinenfunktionen und lassen sich, so wie erprobte mechanische Komponenten immer wieder verwenden. „Der Code wird grafisch aufbereitet und stellt dabei klare Schnittstellen zur Verfügung. Durch diese Strukturiertheit und den Low-Code-Ansatz bleibt Code auch nach Jahren und bei eventuellem Mitarbeiterwechsel lesbar und einfach anpassbar. Code wird nachhaltig und kann übersichtlich weitergetragen werden. Das spart neben Kosten auch Zeit und Nerven. „Meist ändern sich, was die Anwendung selbst betrifft, nur Kleinigkeiten: Evolution statt Revolution. Mit modern umgesetzter Objektorientierung kommen Serienmaschinenbauer einfach zur Software – heute und morgen“, resümiert Aschl. ■



Franz Aschl, Sigmatek

» Wer nicht an
Objektorientierung glaubt,
ist von gestern.

nen abgelesen bzw. konfiguriert werden. Außerdem lässt sich die Hardware-Anordnung im Lasal-Projekt mit der tatsächlichen Konfiguration der Steuerungshardware vergleichen. Eventuelle Unterschiede werden grafisch dargestellt und Produkte können individuell aufgenommen bzw. entfernt werden. Mit komfortablen Tools zur Simulation und Visualisierung aller Prozesse, einer vollständigen Testumgebung bis hin zum Debug-

Direkt zur Übersicht auf
i-need.de
www.i-need.de/p/4225



Ingrid Traintinger,
Leitung Marketing Kommunikation,
Sigmatek GmbH & Co KG
www.sigmatek-automation.com/de