



LASAL von Sigmatek: Low-Code-Entwicklungsplattform mit Objektorientierung und grafischer Darstellung.

OBJEKTORIENTIERUNG IST 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 all-in-one Engineering-Tool LASAL von Sigmatek kommt zudem mit weniger Codes 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, spe-

ziell für Serienmaschinen in unterschiedlichen Ausstattungsvarianten und der Anforderung, diese Maschinen in einen intelligenten Maschinenverbund einzubinden. Das objektorientierte 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. In der OOP werden Code und Daten in logische Einheiten „Objekte“ zusammengefasst, die Maschinenfunktionen entsprechen und über Schnittstellen mit anderen Objekten kooperieren bzw. kommunizieren. Hinter einem Objekt steht jeweils eine Klasse. Sie ist der „Bauplan“ für ein Objekt und definiert den Programmcode und die Datenelemente. Sobald diese Funktionsblöcke mit Parametern und Schnittstellen versehen sind, bezeichnet man sie als Objekte.

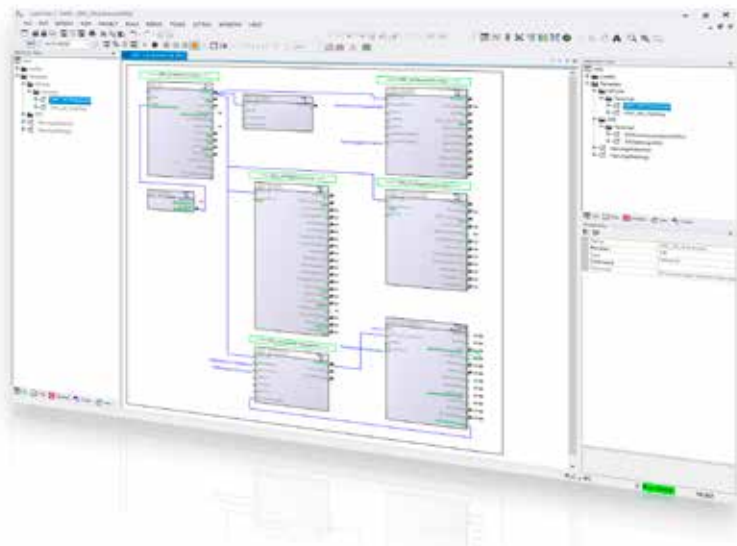
Low Code mit OOP

Bereits seit 20 Jahren setzt der Salzburger Hersteller kompletter Automatisierungslösungen auf Objektorientierung und ist damit der Vorreiter der Branche. Das erprobte all-in-one Engineering-Tool LASAL (IEC 61131-3) ist von Grund auf objektorientiert und zudem als Low-Code-Plattform entwickelt worden. Wie der Name schon sagt, steht Low-Code-Entwicklungsplattform für weniger Codes und weniger programmieren: Software wird „unter Verwendung visueller Applikationsdesigner und anderer grafischer Modellierungsmethoden“ erstellt. Wenn wie bei LASAL die Objektorientierung mit grafischer Darstellung umgesetzt wird, steigen Übersichtlichkeit und Wartbarkeit des Software-Projektes enorm.

Und das ist ein entscheidender 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.

Top-down designen

OOP eröffnet dem Maschinenbauer alle Möglichkeiten und neue 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 es darum, Aufgaben zu zerlegen und visuell darzustellen: 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 vielfältigen 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 Multi-Achsensteuerung oder Deltaroboter-Inbetriebnahme.

„In letzter Zeit ist speziell die für die Kommunikation zuständige Bibliothek stark gewachsen und reicht von TCP/IP über OPC UA Client sowie Server und MQTT hin zu SSL-Verschlüsselung“, weist Franz Aschl auf den vermehrten Bedarf an Vernetzung hin – horizontal >>

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. Der Anwender sieht nicht nur die Daten, sondern auch die Funktionen eines Objektes bzw. einer Klasse.



Bei LASAL ist ein visuelles Live-Debugging möglich, d. h. die aktuellen Werte sind nicht nur im Programmcode, sondern sofort und live in der grafischen Objektabbildung sichtbar. Das spart enorm viel Zeit.

Franz Aschl, Innovationsmanager bei Sigmatek

beispielsweise 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 % reduzieren und der Applikationsingenieur kann sich auf die Implementierung maschinenspezifischer, innovativer Features konzentrieren.

Maximale Transparenz

LASAL ist auch ideal für agile Entwicklungsteams geeignet, da die Arbeit sehr strukturiert erfolgen muss. „Jeder Programmierer kennt das Dilemma von unsauber implementierten Programmen, wo Variable kreuz und quer im Projekt verteilt beschrieben werden, sodass die Auswirkungen einer Programmänderung praktisch nicht vorhersehbar sind“, weiß Franz Aschl. Ganz anders bei der OOP: Die Objekte werden nach außen hin abgeschottet, also „gekapselt“, sodass eine ungewollte Beeinflussung nahezu ausgeschlossen werden kann. „Variable sind nur über zugehörige Methoden manipulierbar. Somit sind klare Schnittstellen vorgegeben, die es dann auch zu verwenden gilt“, so Franz Aschl. Auf die Frage, was bei LASAL anders ist als bei anderen objektorientierten Entwicklungsumgebungen, zeigt sich der Innovationsmanager stolz: „Die visuelle Darstellung der gekapselten Objekte im Netzwerk mit allen Kommunikationsschnittstellen ist einzigartig und sorgt für maximale Transparenz und eben auch das Low-Code-Prinzip.“

Der Code kommt zuletzt

„Durch die grafische Darstellung von Objekten wird der Texteditor anders als bei anderen objektorientierten Tools wirklich erst am Schluss eingesetzt“, erklärt Franz Aschl. Die Implementierung der Methoden in der Klasse kann der Softwareingenieur in den bewährten Programmiersprachen wie Structured Text (ST), Anweisungsliste (AWL), Kontaktplan (KOP) nach IEC 61131-3 oder C vornehmen. Moderne Engineering-Tools wie LASAL machen dem Anwender die Nutzung der objektorientierten Programmierung so einfach wie möglich, indem sie immer wiederkehrende Funktionen übernehmen. Beispielsweise wird der Code für die Deklarationen von Klassen, Variablen, Schnittstellen etc. automatisch im Hintergrund erstellt. „Variable 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“, weist Franz Aschl auf die Effizienz des Tools hin.

Visuelles Live-Debugging

Das realitätsgetreue Verhalten der Komponenten im LASAL-Projekt ist im Online-Modus in Echtzeit zu sehen. In der „Detail-View“ sind mögliche Steckplätze und Busverbindungen ersichtlich, aktuelle Werte und Eigenschaften der Komponenten können abgelesen bzw. konfiguriert werden. Außerdem lässt sich die



Hardware-Anordnung im LASAL-Projekt mit der tatsächlichen Steuerungshardware-Konfiguration 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 Debugger samt zentraler Verwaltung von Projekten und Versionen lassen 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. Das spart enorm viel Zeit“, betont Franz Aschl.

Evolution statt Revolution

„Wer nicht an Objektorientierung glaubt, ist von gestern“, meint Franz 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 der Code auch nach Jahren und bei eventuellem Mitarbeiterwechsel lesbar und einfach anpassbar. Der 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 – für heute und morgen“, ist Franz Aschl überzeugt.

Über 10.000 Klassen umfassen die vielfältigen LASAL-Bibliotheken, zudem stehen moderne Templates sowie einsatzbereite, getestete Softwarekomponenten bereit: **funktions-spezifische Add-ons und themenbezogene Packages.**

www.sigmatek-automation.com