Object-Oriented Into The Future

by Ingrid Traintinger

Pictures: Sigmatek

At the Hanover Fair 2002, SIGMATEK presented the LASAL software platform with object-oriented programming (OOP). As the first automation solution provider, the company said goodbye to strictly sequential programming and has since then consistently relied on object-oriented programming.

To make the application software as universal as possible and at the same time easy to maintain and expand, modern tools and approaches such as object-oriented programming (OOP) are essential for the development of future-proof machine concepts. With an object-oriented approach, machine builders achieve maximum hardware independence. In addition, the software is very clear and structured. Object-oriented programming with LASAL was introduced to the market by SIGMATEK 21 years ago. The idea was to create a single software tool, which, designed as a modular system, takes over all engineering tasks of automation technology – from control programming to project planning of visualization, drive- as well as safety-technology. SIGMATEK was the first manufacturer in this industry to transfer object orientation from PC-programming technology to control technology. "The idea of object-oriented programming (OOP) is to combine code and data into logical units called 'objects'. These objects represent the various parts of a machine or system," explains Franz Aschl, Technology Management at SIGMATEK. Behind each 'object' is a 'class'. It is the blueprint of the object, contains the program code and the associated data elements. Each class can perform a specific 'task', such as measuring and evaluating a temperature, controlling a valve or driving a conveyor belt. Since each individual function is encapsulated, any unintentional external influence can be ruled out. As soon as the class is provided with parameters and interfaces, they are called objects.

"The advantage of this development method is its consistent modularity from the lowest level of the individual function up to the overall project. This ensures clarity and forces structured software development. The dreaded 'spaghetti code' cannot arise at all," says Aschl. Inside, the classes contain the familiar programming via Structured Text (ST), Sequential Function Chart (SFC) or Ladder Diagram (LD) according to IEC 61131-3, as well as ANSI-C and interpreters.

Web-Tipp

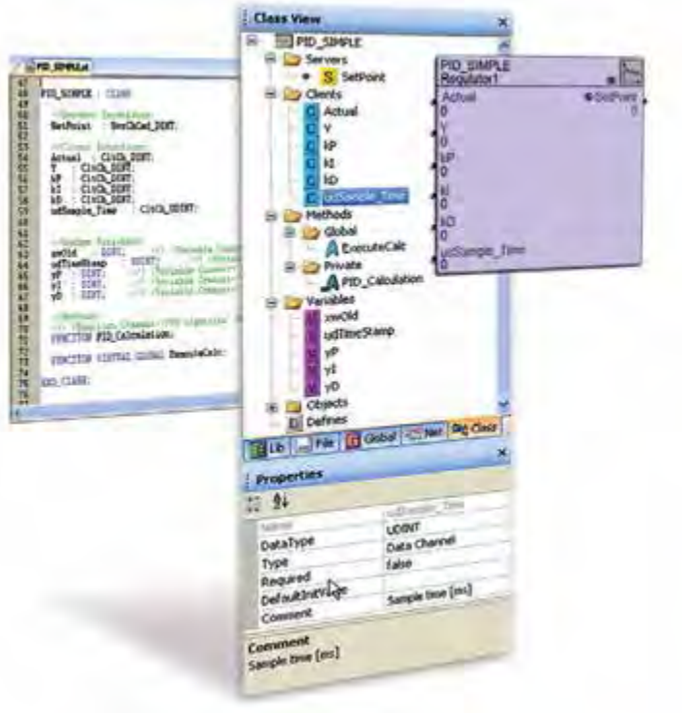
You can get a comprehensive overview of the functional range of LASAL in the following video

youtu.be/7zLRf5ZyW10

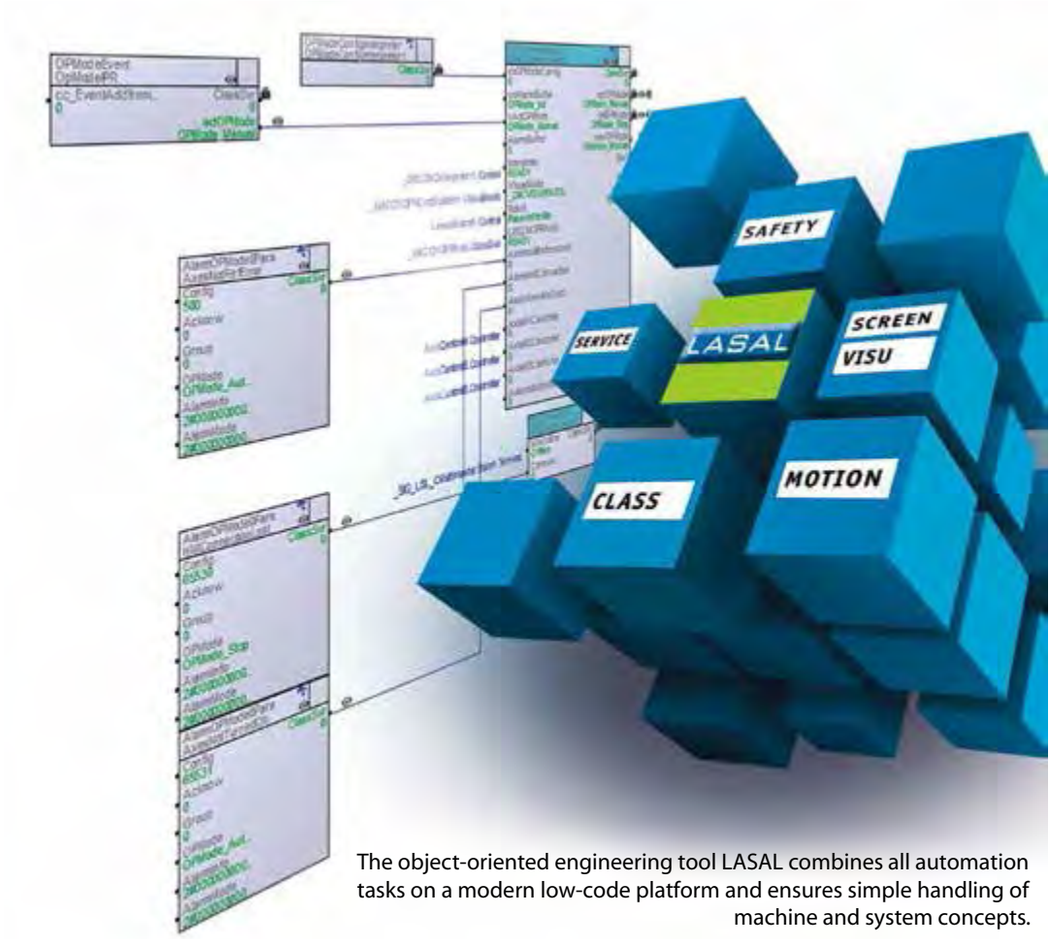
An essential feature of LASAL is the object-oriented programming with client-server technology as well as graphical display, graphical project planning and graphical debugging. At a glance, one receives an overall view of the project, the functionalities, the data traffic and the interfaces. Complex interrelationships are thus much easier to recognize, change and control.
The classes are stored in 'libraries'. Individual functions can be assembled in the software in a modular system and simply 'wired'. As in mechanics, where a tried and tested design is reused, in OOP, thanks to the modular structure, application parts that have been created once can be easily reused and adapted. This ensures a high degree of flexibility, because changes can also be implemented quickly with little code and little effort. This increased reusability significantly reduces the development time and thus the time-to-market cycles.

Continuous Development - from Concept to Detail

"One advantage of object-oriented programming is that it forces thoughtful formulation of definitions," finds Aschl. Starting with the overall project, software architects consider which functions are needed in the machine and how they are interconnected. In top-down design, tasks or functions are broken down and visually represented: Which machine parts talk to others, which exchange data. "It doesn't take a line of code to understand how the application works in a LASAL network. Once the functions and communication interfaces are defined, development begins bottom-up," Aschl explains.
Development tasks can thus be distributed among developers or groups. These tasks can be processed independently. Instead of programming, the user models standard machine functions on the graphical user interface by dragging and dropping from the extensive libraries. This reduces the development effort enormously and increases the software quality at the same time. Only at a certain point is there a need to merge individual software packages into one release and



The class with its properties, methods and interfaces is defined via the class tree of LASAL CLASS. The necessary declaration and graphical representation is done automatically by the engineering tool from Sigmatek.



The object-oriented engineering tool LASAL combines all automation tasks on a modern low-code platform and ensures simple handling of machine and system concepts.

coordinate them with each other. Interface problems are a thing of the past.
Another advantage of object-oriented development is that the hardware does not yet have to be ready or defined. The software is decoupled from the hardware. This is very helpful for the machine builder, since it is often only during development that it becomes clear which computing power, drive type, visualization power and display size are ultimately optimal. Speaking of visualization: A maximum of freedom in the selection of the target device is offered by HTML5 visualizations, which can be flexibly adapted to the HMI used.

Interview with Franz Aschl

What tipped the scales in 2000 for the switch to an object-oriented approach?
We are always in close contact with our customers and when there was an increasing demand for simple engineering for machines with different variants and options, we looked for the right solution. A solution that makes software maintenance easier, provides transparency and supports variant diversity, i.e. that adding options does not affect the rest of the program. All this and much more – namely client-server technology, mechatronic engineering and low code – is offered by object-oriented programming with LASAL.

Looking back, what are you proud of?
That we already recognized the signs of the times back then and were the first company to transfer OOP to automation technology. Of course, we had to convince experienced programmers who were used to ladder diagrams and instruction lists to switch to object-oriented programming. One can say it was a generation break. The younger software engineers already knew object orientation, structured text and C through their training or studies.

Franz Aschl,
Management Technology at Sigmatek in Lamprechtshausen, Austria.

What is special about object-oriented programming with LASAL?
The special feature of LASAL is that, in addition to object orientation, we have relied on client-server technology for communication between the objects from the beginning. This creates clear interfaces and stands for efficient distribution and saving of CPU resources.

To what extent has LASAL changed over the past 20 years?
LASAL was developed as an all-in-one tool: Control, visualization and motion were there from the beginning. Over the years, many comfort functions have been added such as version management, Matlab, hardware editor and scripting. In 2009, the LASAL SAFETYDesigner was added, a seamlessly integrated tool with which users can comfortably configure safety-relevant applications. And since 2015, the LASAL Machine Manager has simplified the handling of multi-CPU solutions. At the heart of this are our extensive libraries, which have grown enormously over the last 20 years and provide templates, add-ons and theme-specific packages, for a variety of machine functions. Of course, we have also integrated HTML5 visualization into LASAL. A new feature is the integration of model-based development with Digital Twin.

Another advantage of object-oriented development is that the hardware does not yet have to be ready or defined. The software is decoupled from the hardware. This is very helpful for the machine builder, since it is often only during development that it becomes clear which computing power, drive type, visualization power and display size are ultimately optimal. Speaking of visualization: A maximum of freedom in the selection of the target device is offered by HTML5 visualizations, which can be flexibly adapted to the HMI used.

Machine Manager: Flexible Task Distribution

In today's world, production plants must be adaptable and able to react quickly and easily to market changes. On the way to the Smart Factory, SIGMATEK has therefore been relying on modular, decentralized automation solutions for many years. Multi-CPU solutions provide machine builders with a great deal of flexibility: The system configuration can thus be put together individually and, if necessary, expanded later with optional functional units and computing power. The 'LASAL Machine Manager' ensures that this intelligent modularity is also represented in the software. Machine software consists of many subprojects, which in turn are subdivided into subgroups such as sequence and visualization. The Machine Manager provides the user with an overview of the individual projects in the machine or plant and allows him to easily see or define: Who is allowed to exchange which data with whom? As a software layer, the Machine Manager lies above the actual machine programming and assigns the various subtasks to the existing hardware control units. This can be defined once during commissioning or changed during the service life of a machine. The communication paths between control units and HMIs can be defined in the Machine Manager. Variables are automatically requested and sent during runtime via cross-project client/server connections. The Machine Manager thus ensures maximum clarity for multi-CPU applications and a great deal of flexibility in the distribution of tasks.

Review: The introduction of the object-oriented, all-in-one automation software LASAL took place at the Hanover fair in 2002.

Engineering mit Zukunft

"Every mechanical engineer knows that the software and the knowledge about it is his greatest asset," Aschl is convinced. In combination with modern hardware, the decisive added value for users is created. With a flexible, structured and user-friendly engineering tool, code remains readable and easily adaptable even after years. Code thus becomes sustainable and can be passed on clearly. Even technology changes, which will inevitably occur more and more frequently in today's world, can be handled with minimal effort and risk.

Ingrid Traintinger
is Head of Marketing at Sigmatek in Lamprechtshausen, Austria.