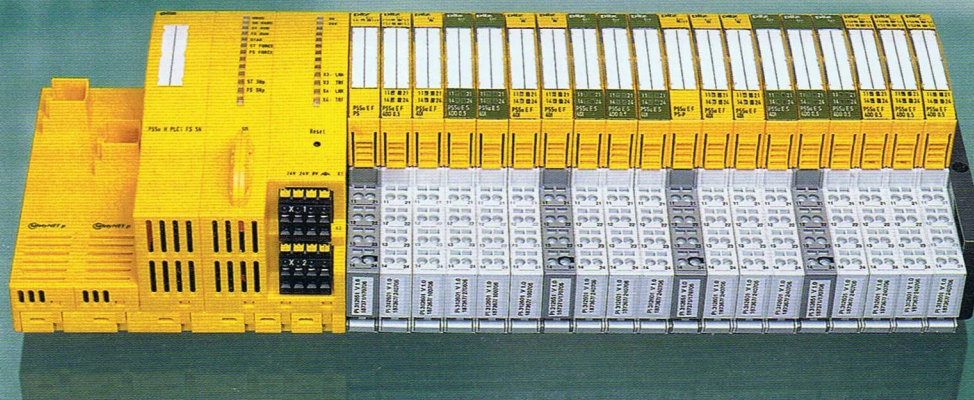
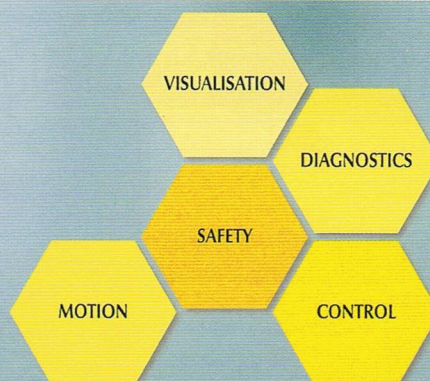


- Met PSS4000 lijkt de besturingscirkel van Pilz weer rond te zijn. Niet langer staat alleen een veiligheidscircuit in de schijnwerper maar een volledig automatiseringssysteem waarin veiligheid, motion, visualisatie, diagnostiek en besturing zijn geïntegreerd.



Automatiseerders brengen grotere flexibiliteit voor PLC's

Meer objectgeoriënteerd programmeren in mechatronica

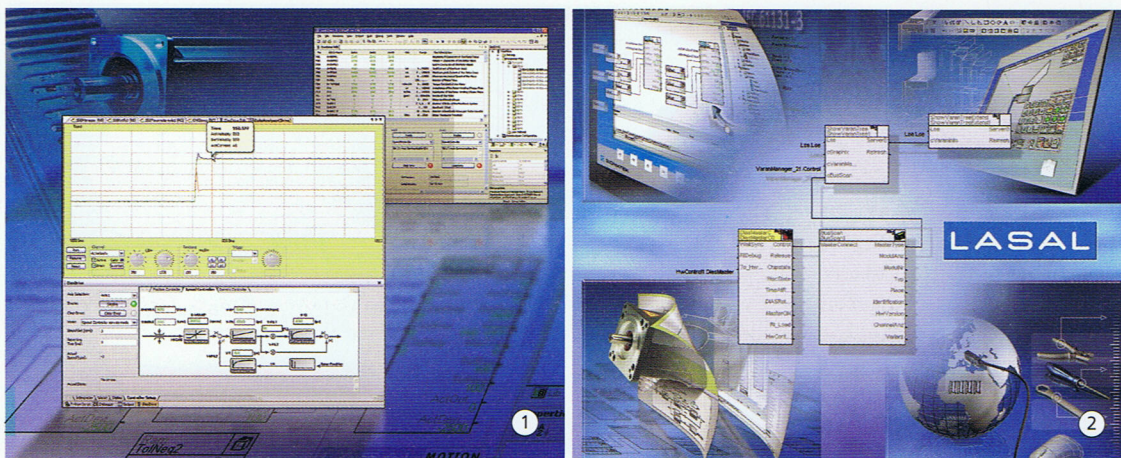
Twee persconferenties met een vergelijkbare boodschap. Aan de vooravond van de Hannover Messe presenteert het Duitse Pilz een nieuw geïntegreerd besturingsconcept, PSS4000, waarin objectgeoriënteerd programmeren een centrale rol vervult. Enkele weken later doet Sigmatek in Oostenrijk de mogelijkheden van het programma Lasal uit de doeken. Tijd om eens wat dieper in te gaan op wat objectgeoriënteerd programmeren is, en hoe beide automatiseerders het inzetten om het leven van de mechatronicus te vergemakkelijken.

• Liam van Koert

Flexibiliteit en schaalbaarheid zijn al jaren toverwoorden in automatiseringsland. Modulariteit wordt hierbij vaak als de oplossing aangedragen. Toch is echte flexibiliteit en schaalbaarheid niet zo vanzelfsprekend als het aankomt op het

programmeren van besturingen en bijbehorende PLC's. Want ondanks het feit dat IEC 61131, de norm voor het programmeren van PLC's, een objectgeoriënteerde benadering niet in de weg staat, blijkt de industriële software in veel gevallen

nog steeds hardwareafhankelijk. Niet alleen moeten software en hardware van dezelfde leverancier zijn, maar ook moet een PLC-programma vaak herschreven worden als de PLC vervangen wordt door een ander type van hetzelfde merk. Met de steeds uitgebreidere functionaliteit die we van onze PLC's verwachten – denk aan motion-control, visualisatie, datalogging en alarmering – wordt het een tijdrovende, onoverzichtelijke en in veel gevallen onmogelijke taak eerdere sequentieel geprogrammeerde codes om te bouwen voor een nieuwe, net iets andere applicatie. Vaak wordt er dan ook voor gekozen maar van voor af aan te beginnen, wat onnodig is en daarmee zonde van tijd en geld. In een tijd waar de



- 1. Bij Lasal Motion zijn zowel regelaars als oscilloscoop in een scherm te bekijken.
- 2. Lasal, dat in 2000 voor het eerst op de markt kwam, bestaat tegenwoordig uit vier delen: Lasal Class, Lasal Motion, Lasal Screen en Lasal Service Tools.

PLC en de IPC steeds meer naar elkaar toe groeien, ligt het voor de hand technieken die hun sporen in de automatisering hebben verdiend, ook in te zetten voor het snel en foutloos optuigen van een machinebesturing. Het leek dan ook een kwestie van tijd voordat een pakket als Lasal van het relatief jonge Sigmatek ook door het grotere publiek bemind werd en een routinier als Pilz het tot de basis maakte van een nieuw volledig geïntegreerd besturingsconcept.

Parallel overzicht

Objectgeoriënteerd programmeren, in de programmeursmond vaak OO ('object oriented') geheten, maakt – hoe kan het ook anders – gebruik van objecten en klassen. De methode die in de jaren zestig van de twintigste eeuw voor het eerst in de programmeertaal Simula te vinden was, kwam dertig jaar later pas echt in zwang. Inmiddels maken vele moderne programmeertalen als C#, C++, Java en Delphi er gebruik van. Bij OO worden in de eerste plaats klassen en objecten gedefinieerd. Klassen kunnen gezien worden als een soort blauwdruk voor de objecten, waarin variabelen, procedures en functies (ook wel methodes genoemd) zijn vastgelegd. Aan een object, dat tot een bepaalde klasse behoort, kunnen vervolgens aanvullende eigenschappen worden toegevoegd. Een klasse zou bijvoorbeeld een gitaar kunnen zijn: een instrument met een body, hals en snaren, die in trilling worden gebracht om muziek mee te maken. Een object dat tot die klasse behoort zou vervolgens een Gibson Les Paul kunnen zijn: een elektrische gitaar met zes snaren, twee humbuckers en een

specifieke vorm die zorgt voor een warm en donker geluid. Objecten en klassen maken vervolgens gebruik van drie pijlers: inkapseling, overerving en polymorfie. Inkapseling houdt in dat een object precies die functie vervult waar het voor bedoeld is, ongeacht waar de input vandaan komt, en ongeacht waar de output naar toe gaat. Een Gibson Les Paul is om muziek mee te maken en niet om ramen mee te zemen. Daarbij maakt het Les Paul niet uit of de snaren met plectrum of tanden in trilling worden gezet, en of er een koptelefoon of zware Marshall-versterker aan de uitgang hangt. Objecten vormen bouwstenen waar je afzonderlijk weinig mee aanvangt, maar die samen een goedwerkend systeem vormen. Bij overerving wordt er met basisklassen en subklassen gewerkt. Je gebruikt dit principe wanneer je meerdere klassen hebt die erg op elkaar lijken maar net iets afwijken. Nu kun je de ene kopiëren, plakken en iets aanpassen. Maar met de veelvoorkomende zelfde stukjes code loop je nu het risico dat wanneer je iets moet aanpassen, je het ook in alle kopieën moet aanpassen. Het is dus beter een hiërarchie op te bouwen, waarbij je de gemeenschappelijke kenmerken in een klasse onderbrengt en de aanvullende subklassen met specifieke eigenschappen definieert.

Tot slot is er polymorfie. Dankzij deze eigenschap van objectgeoriënteerd programmeren is het mogelijk al in een vroeg stadium een functie voor generieke objecten te definiëren, die later op specifieke objecten kan worden losgelaten, zonder dat de functie van het bestaan van de specifieke objecten weet. In het voorbeeld van de gitaar: een

functie 'E aanslaan' zal voor de Gibson resulteren in een toon met een frequentie van 82,4 Hz, terwijl dit voor een basgitaar 41,2 Hz zal zijn.

Tot zover de theorie. Wat zijn nu eigenlijk de voordelen voor een mechatronicus van een dergelijke werkwijze? Op de eerste plaats gebeurt het programmeren gestructureerder, waarmee het allemaal een stuk overzichtelijker wordt. Het denken in objecten is eenvoudiger dan het denken in ingewikkelde code. Dit betekent in de regel minder fouten en een snellere en flexibelere manier van werken. Objecten en klassen kunnen gemakkelijk worden aangemaakt, getest, worden vrijgegeven, in een bibliotheek worden opgenomen, om vervolgens te worden hergebruikt of aangepast. Met de huidige visualisatiemethoden doen de ingewikkelde stukjes code waaruit ze zijn opgebouwd onder het oppervlak hun werk. Daarnaast kan volgens goed mechatronisch gebruik de engineering werkelijk parallel plaatsvinden. Terwijl voorheen eerst het mechanisch ontwerp plaatsvond, gevolgd door het elektrisch ontwerp en daarna pas de programmering, hoeft de specifieke hardware pas nu in een veel later stadium gekozen te worden, wat resulteert in veel kortere ontwikkelingstijden.

PAS4000 en PSS4000

Velen kennen Pilz als leverancier van veiligheidsoplossingen. Het geel van de huisstijl en de verschillende componenten zou je dan ook bijna doen vergeten dat het bedrijf zestig jaar geleden begon met 'gewone' recht-toe-rechtaan-besturingen. Met de komst van een objectgeoriënteerde benadering lijkt

de besturingscirkel weer rond te zijn. Hoewel standaard besturingscomponenten evenals sensoren en actuatoren het productportfolio nooit hebben verlaten, knoopt een nieuwe Automation Suite dat onder de naam PAS4000 door het leven gaat, alle elementen aan elkaar. De objectgeoriënteerde benadering, waarbij er geen hoofdprogramma bestaat dat de subroutines sequentieel start, maakt werkelijk modulair werken mogelijk. Objecten en subprogramma's draaien tegelijkertijd met elk een eigen tijdcyclus (instelbaar van 1 tot 100 milliseconde) en een eigen prioriteit. De hardwareafhankelijkheid maakt wijzigen, opschalen of downsize zeer eenvoudig. Terwijl de machinebouwer voor zijn machines met weinig I/O meestal koos voor een overgedimensioneerd systeem om zo een standaardbesturing voor al zijn machines te kunnen realiseren, is maatwerk nu geen probleem, zonder aan standaardisering en ontwikkelingstijd in te boeten. De komst van de nieuwe Automation Suite van Pilz staat niet op zichzelf. Het maakt deel uit van een totaal besturingsconcept, PSS4000 genaamd, waarin ook veiligheid, motion-control, diagnose en visualisering volledig geïntegreerd zijn. Sinds augustus is er nieuwe hardware op de markt, met PAS4000 als spin in het web. Hierbij worden drie niveaus onderscheiden. Ten eerste is er een kopmodule die voor de afhandeling van alle communicatie zorgt. Aan deze kopmodules kunnen vervolgens maximaal 64 basismodules gekoppeld worden, die op hun beurt aan specifieke modules gekoppeld worden en daarmee fungeren als een schaalbare I/O-interface. Het geheel communiceert hierbij via het op Ethernet gebaseerde SafetyNETp. Sinds kort zijn er drie varianten van het PSS4000-systeem op de markt verkrijgbaar. Op de eerste plaats is er de PSS Universal PLC-serie. Deze PLC's zijn gewoon (via PAS4000) programmeerbaar met ladderdiagrammen of gestructureerde tekst, zoals dit in IEC 61131-3 is vastgelegd. De PNOZmulti-programmering is hier uitgebreid met complexere standaard- en veiligheidstaken. Een tweede aanwinst is de PSSUniversal Multi-serie. Deze serie borduurt voort op het PNOZmulti-systeem. Veiligheidsfuncties kunnen via parametrisering geprogrammeerd worden. De laatste op de markt verschenen serie betreft PSSUniversal I/O-hardware. De serie bestaat uit een groot aantal stan-

De specifieke hardware hoeft nu pas in een veel later stadium worden gekozen

daard en failsafe I/O-modules die voor decentrale toepassingen kunnen worden ingezet.

De onderdelen motion-control en visualisatie staan voor PSS4000 nog op de rol. Wel is de configuratietool voor PNOZmulti-systemen nu de grafische programmeertaal, die als PASmulti-editor in PAS4000 beschikbaar is.

Lasal en DIAS

Binnen twaalf jaar na de oprichting in 1988 komt het Oostenrijkse Sigmatek met een eerste versie van zijn objectgeoriënteerde automatiseringspakket. Lasal is daarmee een van de eerste tools die de flexibele OO-werkwijze binnen het PLC-domein brengt. Ergens is het niet verwonderlijk dat juist dit bedrijf software ontwikkelde die gestoeld is op praktijken uit de wereld van de pc: de besturingscomponenten zijn met het takenpakket, dat naast besturing ook visualisering omvat, meer IPC's dan PLC's. Lasal heeft de afgelopen jaren wel wat veranderingen ondergaan. Zo bestaat het tegenwoordig uit vier onderdelen, die elk hun eigen toepassing kennen. Het eerste onderdeel is Lasal Class. Hiermee vindt het daadwerkelijk programmeren volgens IEC 61131-3 plaats. Dit kan onder meer gebeuren met 'structured text', ladderdiagrammen of C. Andere functies zijn directe I/O-toegang, debugging en het online aanbrengen van wijzigingen. Ook Lasal Class maakt gebruik van een vereenvoudigde grafische weergave die de daadwerkelijke code verstopt. Een tweede onderdeel is Lasal Screen. Deze tool is speciaal ontwikkeld voor de visualisatie van een applicatie. Hierbij zijn, naast tools voor tekenen en uitlijnen, tal van standaardvoorgedefinieerde functies als alarmmanagement, datalogging en receptmanagement aanwezig. Het derde onderdeel is Lasal Motion, dat bedoeld is voor de regeling en optimalisatie van aandrijvingen. Het is geschikt voor complexe en precieze aansturingen, waarbij regelaars en oscilloscoop in hetzelfde scherm te zien zijn. Het laatste onderdeel betreft Lasal Service Tools. Dit bevat tools voor onderhoud op afstand,

software-updates en gegevensuitwisseling, wat onder meer kan plaatsvinden via een remote manager, OPC-server of webserver.

Lasal staat niet op zichzelf. Hoewel het geen probleem is het in combinatie met andere hardware te gebruiken, heeft Sigmatek, dat in Nederland door het Barendrechtse SigmaControl wordt vertegenwoordigd, het Distributed Automation System in het portfolio. Deze familie van I/O-systemen, die zowel in standaardvorm (DIAS), compacte vorm (C-DIAS), veilige vorm (C-DIAS safety) als extra beschermde vorm (P-DIAS) voorkomt, maakt gebruik van de zeer snelle op Ethernet gebaseerde Varan-bus, die cyclustijden tot onder 100 microseconde kent. Dit geldt ook voor de diverse frequentieregelaars, HMI's, CPU's en IPC's die het aanbod voltooien.

Tot slot

In de wereld van de pc, maar ook in de embedded wereld, worden al jaren de vruchten van objectgeoriënteerd programmeren geplukt. En hoewel de industriële automatisering zich grotendeels lang afzijdig heeft gehouden, lijken nu toch de eerste stappen naar acceptatie gezet te worden. Met het vergroen van werelden die embedded partijen als National Instruments (LabView) ook richting motion-control doen verschuiven en de pc die steeds verder het automatiseringsdomein binnendringt (wat vonden we een aantal jaren terug ook al weer van Ethernet?), kon het ook niet uitblijven. Wel zal er bij automatiseerders nog wat kennis moeten worden bijgespijkerd en zullen oude gewoonten moeten worden afgeleerd. Zowel Pilz als SigmaControl is maar al te graag bereid hier in de vorm van een gratis seminar of workshop aan mee te helpen. •

Liam van Koert is freelance auteur