www.elektroniknet.de

Elektronik

FACHZEITSCHRIFT FÜR INDUSTRIELLE ANWENDER UND ENTWICKLER

27. November 2001 | 24 | 9.- DM 67.- S 9.- sfr.

S 44
Grafische Programmierung in der Automation: Ein neues, grafisches Tool überspringt das Stadium des Quellcodes und eröffnet der Automatisierungswelt überraschende Perspektiven.

Automatisieren ohne Code

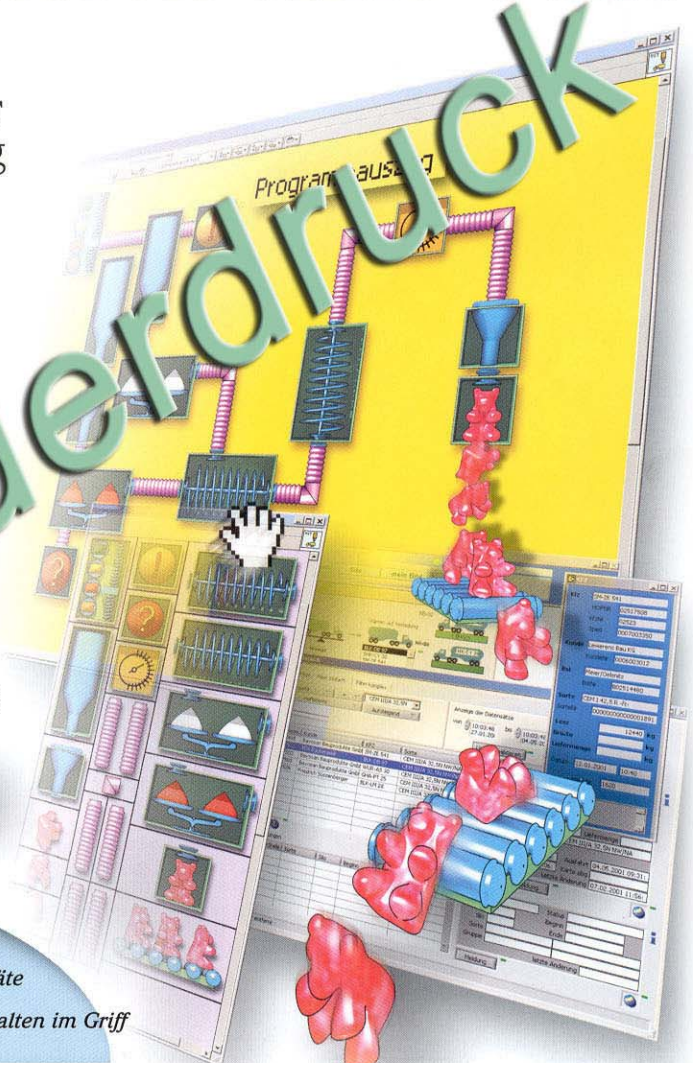
S 54
IEC 61499 und Profinet
Gemeinsame Sprache für verteilte Automatisierung

S 66
Bluetooth-Serie, Teil 7
Angepasste Protokolle nach OBEX-Standard

S 74
Motorsteuerungen
Intelligente Treiber und Controller optimieren Motorverhalten

S 84
Sensorik und Aktorik
Universell einsetzbares CAN-Bus-Sensorinterface

Besuchen Sie die „Elektronik" auf der SPS/IPC/DRIVES in Halle 4, Stand 112



Embedded

In diesem Heft:
S 94 **Spezial „Embedded Design"**
- *Entwicklungstool: IEEE-1394-Geräte durchleuchtet*
- *Prioritätssteuerung: Echtzeit-Verhalten im Griff*
- *Neue Embedded-Produkte*

Automatisieren ohne Code

Grafische Programmierung erobert die Automatisierungswelt

Die Nutzung der Objektorientierung mit Ereignisbehandlung, Vererbung etc. konnte sich in der Automatisierungstechnik nicht so recht durchsetzen – nicht zuletzt deshalb, weil objektorientierte Sprachen kompliziert sind und eine neue Herangehensweise gegenüber prozeduralem Programmieren erfordern. Ein neues, grafisches Tool überspringt das Stadium des Quellcodes und öffnet der Automatisierungstechnik damit völlig neue Möglichkeiten.

Von Herbert Pichlik und Hilde-Josephine Post

Komplexe Automatisierungslösungen mit verschiedenen Ebenen und vielen verteilten intelligenten Systemen besitzen ihre Tücken. Sie stellen hohe Anforderungen an die softwaretechnische Seite. Noch muss sich der Applikationsingenieur mit zahlreichen Programmierwerkzeugen herumschlagen. Schnittstellenprobleme, unzureichende Transparenz und mangelnde Handhabbarkeit sind an der Tagesordnung. Einem Pionierunternehmen aus Jena gelang scheinbar der Clou: Es setzt die grafische Entwurfsphilosophie direkt in Software-Programmierung und Visualisierung um. Ereignisgesteuerte Objektnetze mit Vererbungsstrategien versprechen enorme Kosten- und Zeitersparnisse (*Bild 1*).

„In der Vergangenheit benutzten wir als Automatisierungsausrüster für die Visualisierung unterschiedlichste Tools, etwa die von Siemens oder Wonderware“, berichtet Jens Vogel, Geschäftsführer der Vogel Automatisierungstechnik GmbH. „Sie reichten für unsere Funktionsanforderungen oft nicht aus“, so Vogel weiter, „also mussten wir zusätzlich Datenbank-Tools oder etwa LabVIEW von National Instruments und schließlich noch eine Speicherprogrammierbare Steuerung (SPS) hinzunehmen. Das gesamte System enthielt dann vier grundverschiedene Programmieroberflächen. Der Endanwender forderte aber Einheitlichkeit. Großes Kopfzerbrechen bereiteten uns hier die Schnittstellen.“

LabVIEW als Basis

Um das Drama endgültig zu beenden, entschloss sich das Jenaer Unternehmen, eine zukunftsträchtige Lösung zu entwickeln, mit der Applikationsingenieure komplexe Automatisierungssysteme leicht handhaben können. Vogel wollte das Rad aber nicht neu erfinden. Deshalb schaute sich sein Entwickler-Team nach marktgängigen Werkzeugen um, bei denen es sich lohnte, sie so zu ergänzen, dass sie universell einsetzbar sein würden. Die Wahl fiel auf LabVIEW (Laboratory VIrtual Engineering Workbench). Drei Entscheidungskriterien standen dabei im Vordergrund: fortschrittliche Technologie (Datenfluss), ausreichend Geschwindigkeit, freie Programmierbarkeit. „Wir entwickelten Zusatzfunktionen, die uns auf ganz neuem Niveau programmieren lassen“, erklärt Vogel stolz. Dem Pionier ging es darum, auf grafischem Weg bessere softwaretechnische Strukturierungsmöglichkeiten zu schaffen, die vom Entwurf über das Programm bis hin zur Visualisierung und letztlich zur klaren Dokumentation führen.

Bisher war LabVIEW vor allem auf Messaufgaben begrenzt und nicht als allgemeines Programmierwerkzeug großer verteilter Systeme mit vielen parallel laufenden Prozessen gedacht. „Das ändert sich jetzt grundlegend durch unsere zusätzlichen Objekttechnologie-Toolsets für LabVIEW“, ist Vogel überzeugt. Aus LabVIEW und den Objekttechnologie-Toolsets entstand ObjectVIEW. Die Jenaer Entwickler verknüpften grafische Datenflussprogrammierung, wie sie bislang in LabVIEW erfolgreich eingesetzt wird, mit modernen objektorientierten und ereignisgetriebenen Ansätzen. Grafisch ließen sich nun beliebig umfangreiche Applikationen erstellen, so Vogel, die zuvor mühevoll in Java oder C++ codiert werden mussten.

Intelligente, dezentral verteilte Software-Knoten

Prinzipiell setzten die Entwickler softwareseitig auf eine Struktur, die hardwareseitig schon längst in der Automatisierungstechnik Einzug gehalten hat. Sie entwickelten Toolsets, die auf intelligenten, verteilten Software-Knoten basieren. Die Funktion eines realen Objekts, wie ein Stellantrieb mit Sollwertvorgabe, Messwert, Regler zur Ansteuerung der Aktoren und Parameterverwaltung, wird einem Software-Objekt zugeteilt, das in sich die Funktion kapselt. Mit weiteren Modultypen geschieht Gleiches. Dadurch besteht die Applikation nicht mehr aus riesigen, unüberschaubaren Blöcken, sondern aus vielen kleinen Komponenten, die

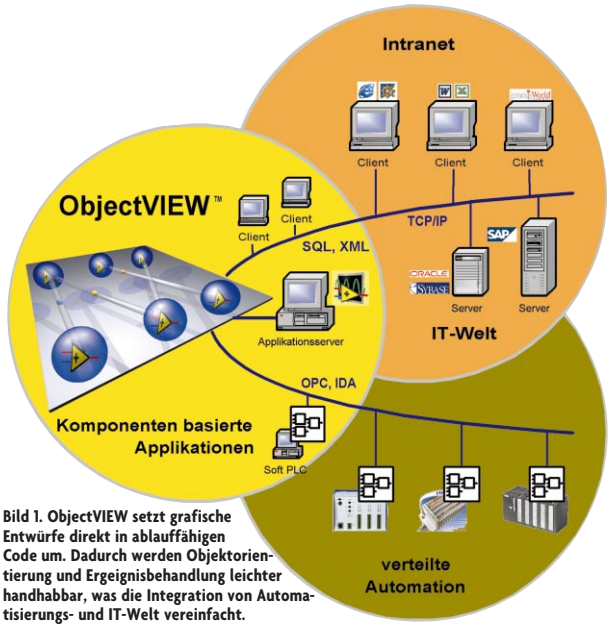


Bild 1. ObjectVIEW setzt grafische Entwürfe direkt in ablauffähigen Code um. Dadurch werden Objektorientierung und Ergeignisbehandlung leichter handhabbar, was die Integration von Automatisierungs- und IT-Welt vereinfacht.

miteinander interagieren. Für eine derartige komponentenbasierte Struktur bietet LabVIEW gegenüber konventionellen, textbasierten Programmiersprachen eine Vielzahl strategischer Vorteile, die laut Vogel jedoch für komplexe Systeme unzureichend waren.

LabVIEW-Programme werden grafisch, und zwar mit Hilfe von Blockdiagrammen und Frontplattenelementen erzeugt (*Bild 2*). Im Fachjargon nennt man das Virtuelle Instrumentierung (VI). In Analogie zu einem realen Messgerät sieht der Anwender nun auf seinem Bildschirm die gleichen Bedien- und Anzeigeelemente. Dieses, in Pixel gegossene Instrumentenbrett dient als User-Interface.

Ein Blockdiagramm (*Bild 3*) repräsentiert den eigentlichen Programm-

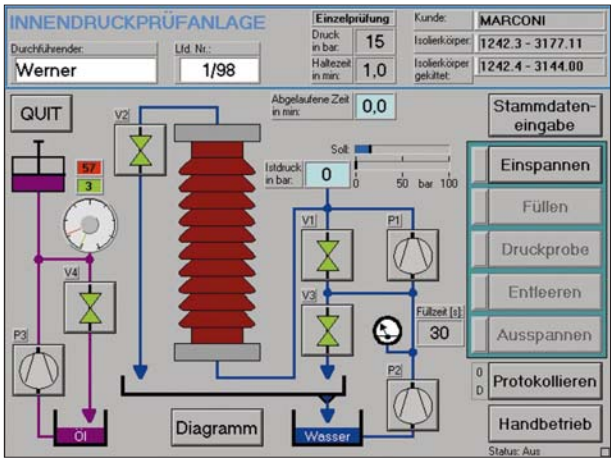


Bild 2. Die Bedienoberfläche eines Programms ist in LabVIEW in Analogie gehalten zu den Anzeigeelementen von Messgeräten. Der Anwender sieht auf dem Bildschirm das Instrumentenbrett. Er kann darüber mit dem Programm in Interaktion treten.

SYSTEC
Gesellschaft für Automatisierung,
Systeme und Dokumentation mbH
Tel. 0911 – 99 89 55 0
Fax. 0911 – 99 89 55 633
Schreiberhauer Str. 5
90475 Nürnberg
www.systec-gmbh.com
info@systec-gmbh.com
Ein Unternehmen der Papp Gruppe
www.papp-gruppe.de

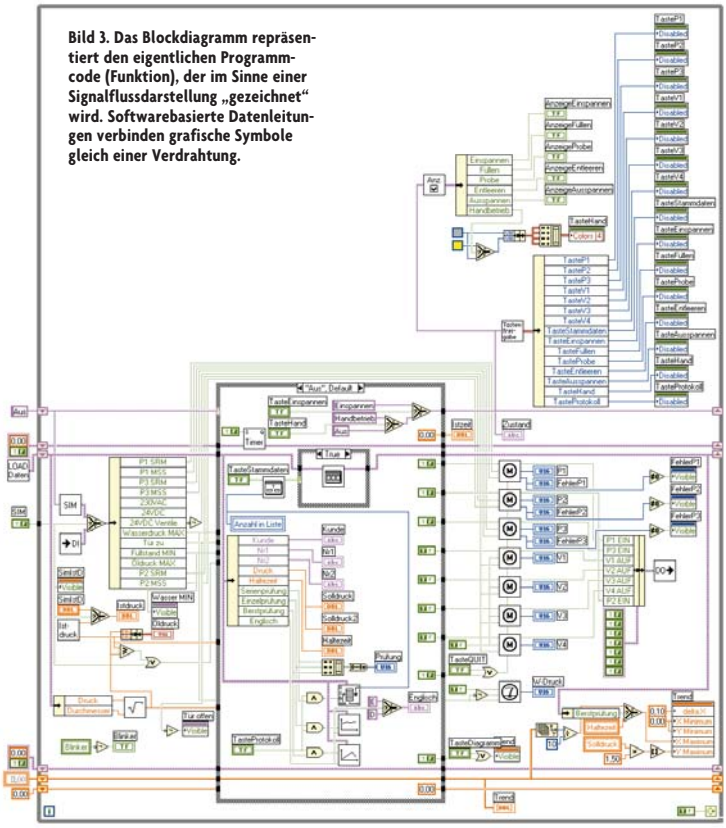


Bild 3. Das Blockdiagramm repräsentiert den eigentlichen Programmcode (Funktion), der im Sinne einer Signalflussdarstellung „gezeichnet" wird. Softwarebasierte Datenleitungen verbinden grafische Symbole gleich einer Verdrahtung.

code, der im Sinne einer Signalflussdarstellung „gezeichnet" wird. Softwarebasierte Datenleitungen verbinden grafische Symbole gleich einer Verdrahtung. Bestandteile des Blockschaltbildes können sowohl einfache mathematische Operatoren wie Addierer, Multiplizierer etc. sein, aber auch komplexe Funktionen wie Fast-Fourier-Transformation (FFT) oder selbst definierte Unterprogramme (SubVI). Mit Hilfe solcher SubVIs lassen sich modulare und hierarchische Strukturen nachbilden. Dabei ist die Anzahl der Hierarchieebenen nicht begrenzt.

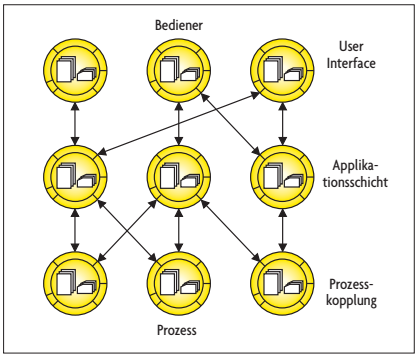


Bild 4. Durch die Objekt-Toolsets ist eine verdeckte Schicht, die Applikationsschicht, hinzugekommen, die nun ausschließlich für die Funktion verantwortlich ist.

Ein weiteres Manko stellte sich bei LabVIEW dar. Auf die Bedienerschnittstelle folgt zugleich das Programm (Blockdiagramm). Es ist nur dann wirksam, wenn der Bediener die Benutzerschnittstelle auf dem Bildschirm aufruft. Ein Motor würde also nur dann laufen, wenn seine Steuerung auf dem Bildschirm sichtbar ist. Bei kleineren Programmen und vor allem bei Messaufgaben sei das ausreichend, erklärt Vogel, bei komplexeren Steuerungsaufgaben aber nicht. Hier müssen Visualisierung und Funktion der Anwendung vollkommen unabhängig voneinander verwaltet werden. Deshalb teilten die Entwickler die Software in verschiedene Schichten ein. Eine verdeckte Schicht, die Applikationsschicht, ist nun ausschließlich für die Funktion verantwortlich (*Bild 4*). Sie befindet sich ständig im Einsatz, unabhängig davon, ob der Bediener sie aufgerufen hat. Die Applikationsschicht besitzt sowohl eine Schnittstelle zum Benutzer als auch zum Prozess.

Aktive Objekte und grafische Vererbungsstrategien reduzieren Programmieraufwand

Was außerdem bei LabVIEW gefehlt hat, war eine grafische Objektorientierung, die mit aktiven Objekten arbeitet (*Bild 5*) und mit der sich Eigenschaften grafisch vererben lassen (Mehrfachvererbung). Es existiert zwar ein GOOP-Ansatz (Graphical Object Oriented Programming), er unterstützt aber lediglich passive Objekte. Passiv heißt: Bei einem Sensor als Objekt legt dann eine Methode den Zustand in einen Datenspeicher ab – z.B., ob der Sensor ein- oder ausgeschaltet ist. Es muss aber jemanden geben, der das Objekt „anschubst". Vogel erklärt: „Wir haben diese Idee erweitert, so dass Objekte nun selbstständig agieren." Nun beherbergen „aktive Objekte" einen oder mehrere Akteure (Objektprozess(e)). In Kombination mit den Objekt-Toolsets sei LabVIEW nun fähig, aktive Objekte auch dynamisch zu erzeugen. Grafische Klassen- und Vererbungsstrategien unterstützen dabei die Aufgabe, komplexe Gebilde besser beschreiben zu können. Man definiert zum Beispiel eine Basisklasse. Das könnte ein Motor mit Ein-/Ausschaltfunktion sein. Sollen jetzt weitere Motorsteuerungen definiert werden, die zusätzliche Eigenschaften benötigen, etwa rechts-/links-drehend sind, so vererbt die Basisklasse ihre Eigenschaften. Ein weiterer Programmbaustein umfasst dann nur jene Eigenschaften, die im Verhalten neu hinzugekommen sind. Auch hiervon lassen sich dann wiederum andere Motorsteuerungen ableiten. So entstehen Vererbungshier-

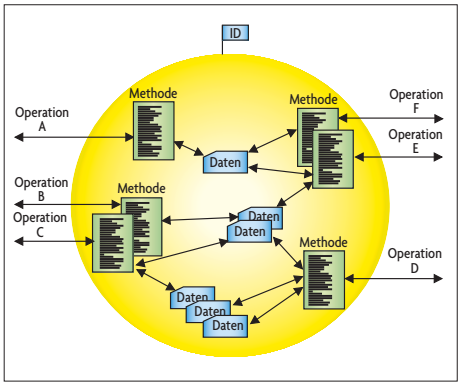


Bild 5. Eine grafische Objektorientierung, die mit aktiven Objekten arbeitet, fehlte bisher in LabVIEW. Sie sind aber zwingend notwendig, um komplexe Prozesse abbilden zu können.

archien. Dieses Vorgehen spare extremen Programmieraufwand und schaffe hervorragende Übersichtlichkeit, so die Entwickler.

Bei der Erstellung von komplexen LabVIEW-Programmen sind Software-Elemente wie Occurencies, Queues, Notifications, Call by Reference Nodes, Semaphoren und Rendezvous notwendig. Damit gingen die grafischen Vorteile von LabVIEW verloren, meint Vogel. Denn diese Kommunikationsbeziehungen seien grafisch nicht sichtbar. Große Programme würden unübersichtlich. Hingegen erlaube ObjectVIEW selbst bei komplexen Applikationen grafische Programmierung, weil es die Software in geschlossene Objekte mit definierten Schnittstellen kapselt. Die Kommunikationsbeziehungen zwischen Objekten lassen sich in der Applikationsschicht grafisch verdrahten. Es müssen nur noch Bausteine ausgewählt und intuitiv zusammengesetzt werden. Der Applikationsingenieur muss nicht mehr wissen, was innen passiert. Er spare sich Übersetzungs- und Verknüpfungsarbeit, die hingegen im klassischen informationstechnischen Bereich nötig sei, wie Vogel bemerkt, und oft hohe Kosten verursache.

Ereignissteuerung zwingend notwendig

Eine weitere wichtige Ergänzung zu LabVIEW verbirgt sich hinter dem Ereignisflussmodell, das mit den Toolsets ins Leben gerufen wurde. Objekte interagieren ereignisgesteuert miteinander. Von der Logik her erinnert die Struktur des Duos LabVIEW/ObjectVIEW an Funktionsbausteine eines SPS-Programms. Die Leistungsfähigkeit ist allerdings um Zehnerpotenzen höher, da es nicht nur um Bits und Bytes geht, sondern auch um Statistiken, Datenbankabfragen, Bildverarbeitung oder komplizierte Graphen und vieles mehr. Hier ist es nicht mehr möglich, alle Funktionsbausteine in einer Schleife zu durchlaufen, wie das bei einem SPS-Programm geschieht. Das würde die Abarbeitungszeit in die Höhe treiben. Da LabVIEW derzeit aber genau auf diese Weise funktioniert – es arbeitet ständig die Schleife ab – lassen sich keine komplexen Vorgänge abbilden. Hier setzt ObjectVIEW mit vielen kleinen

Technische Perspektiven

ObjectVIEW repräsentiert unterschiedliche Designkonzepte, angefangen vom Datenflussansatz, den sich LabVIEW zu eigen macht, über objektorientierte Ansätze mit Klassen und Mehrfachvererbung, Ereignis-, Signal- und Botschaftenmanagement, Adressierung über URLs, verteilten Objekten, Aggregation, Ports, Petri-Netzen und Objektnetzen. Die Objekte sind im lokalen Netz (LAN) und eingeschränkt im Weitverkehrsnetz (WAN) inklusive Internet verteil- und administrierbar. Derzeit existieren vier Varianten von ObjectVIEW:

- *Object-Event-Toolset:* Diese Basisversion unterstützt „aktive Objekte“ und Ereignissteuerung, die auf asynchrone Botschaften (mit Hilfe von Queues) sowie Signale (über Nachrichten) reagiert. Eine grafische Verschaltung der Objekte ist mit dieser Version nicht möglich. Die Adressierung erfolgt über URLs. Kommunikationstechnologien wie ActiveX und OPC/DataSocket finden Unterstützung.
- *Class-Inheritance-Toolset:* Zusätzlich zur Basisversion gibt es „passive Objekte“, Klassenbibliotheken und Klassenbibliotheksmanagement, Mehrfachvererbung, Methoden, Methodenaufrufe sowie flache Zustandsautomaten.
- *Object-Net-Toolset:* aktive Objekte lassen sich grafisch verschalten – hierarchische Objektnetze mit Ports. Der Zustandsfluss in Prozessen ist mit Petri-Netzen programmierbar: Bedienungs-/Ereignisnetze, Stellen-/Transitionsnetze, Colorierte Petri-Netze oder Harel-State-Machines.
- *Application-Framework-Toolset zur Erstellung von Client-Server-Applikationen:* Es nutzt alle Features der untergeordneten Toolsets und bietet unterschiedliche Klassenbibliotheken wie Client/Server, Node-Communication/-Management, Message-Management, Trends, Error-Management, Facility-Management. Enterprise-Resource-Planning und Manufacturing-Execution-Systeme lassen sich damit grafisch programmieren.

Das multiplattformfähige LabVIEW unterstützt Multithreading auf allen wichtigen Betriebssystemen mit unterbrechbarem Multitasking (Windows 2000/98/ME, Linux, Solaris, Concurrent PowerMAX). Mit seinem grafischen Compiler erzeugt LabVIEW sehr schnellen Programmcode, vergleichbar mit C++. Außerdem ist das Tool offen und unterstützt alle relevanten Schnittstellen und Kommunikationsmechanismen. Es gibt Treiber für alle wichtigen Mess- und Automatisierungskomponenten wie OPC-Schnittstelle, TCP/IP, ActiveX , Visual Basic, C, DLLs.
In der Anfangsphase werden die Objekt-Toolsets nur LabVIEW-Schnittstellen unterstützen. In naher Zukunft soll es jedoch auch direkte Objektschnittstellen geben. Die nächste ObjectVIEW-Version enthält eine XML-Schnittstelle. „Dann sind wir auch in der Lage, auf der Business-Ebene und im Internet Daten zu veröffentlichen“, sagt Jens Vogel. Zudem soll das Microsoft Simple Object Access Protocol (SOAP) unterstützt werden. Damit ist die Kommunikation verteilter Objekte auch zu anderen Systemen möglich.

Schleifen und ereignisgesteuerten Methodenaufrufen an. Das heißt, Objekte laufen nur dann los, wenn sich an den Eingängen etwas ändert. Dadurch wird der Objektprozess angestoßen – das Objekt (der Messprozess) wacht auf, ohne zwischenzeitlich Rechenzeit verschwendet zu haben. Zwar existieren auch Systeme, die gepollt werden müssen, etwa Bussysteme. Benutzerschnittstellen lassen sich jedoch völlig ereignisorientiert auslegen; ebenso die in den Toolsets hinzugefügte Applikationsschicht. Da nicht ständig Hintergrundaktivitäten stattfinden, steht im Moment der Aktion eine fast 100-prozentige Rechnerleistung zur Verfügung. Einzelne Aktionen werden enorm verkürzt. Vogel möchte deshalb nochmals betonen: „Ereignissteuerung war zwingend notwendig, um LabVIEW auch in komplexen Systemen einsetzen zu können."

Ein weiterer Vorteil, komplexe nebenläufige Automatisierungsprozesse leicht handhabbar zu machen, verbirgt sich in der Datenflussprogrammierung von LabVIEW. Stehen Multiprozessorsysteme zur Verfügung, so lassen sich die Aufgaben automatisch auf zahlreiche Prozessoren aufteilen. Bei vielen anderen Software-Werkzeugen stellt das noch ein großes Manko dar. Übliche C- oder Java-Programme schaffen das nicht, weil sie von Natur aus sequentiell arbeiten und nur durch aufwendige Programmierung (z.B. mit Hilfe von Threads und/oder Prozessen) parallelisierten Code erzeugen können. Der datenflussbasierte Ansatz sei, so die Entwickler, eine wesentlich elegantere und natürlichere Art, Probleme zu formulieren. So ist es in LabVIEW sehr leicht möglich, konkurrierende, einzeln ablaufende und leicht synchronisierbare Programmteile zu gestalten.

Interview

Chance zum universellen Programmierwerkzeug

Über die Marktbewertung des Zusatz-Tools ObjectVIEW zu LabVIEW sprach die *Elektronik* mit dem Technik- und Marketing-Chef Rahman Jamal von National Instruments.

■ *Das Zusatz-Tool ObjectVIEW der Vogel Automatisierungstechnik GmbH scheint eine neue Generation grafischer Programmierumgebungen einzuläuten. Ist es für Sie eine Konkurrenzentwicklung? Eigentlich hätte das doch auch National Instruments entwickeln können?*

■ **Jamal:** Für uns ist es eher ein Kompliment, wenn einer unserer Allianzpartner neue Wege geht, die bisher noch nicht beschritten wurden – etwa die Entwicklung eines objektorientierten Konzepts mit der grafischen Methodik von LabVIEW. Es ist ein Beweis für die hohe Flexibilität und Offenheit von LabVIEW, dem wir letztlich unseren Erfolg verdanken. Deshalb sehe ich es als eine gute Ergänzung an und nicht als Konkurrenz.

■ *Bisher war LabVIEW hauptsächlich für Messaufgaben gedacht. Kann es durch die Zusatzentwicklung zu einem universellen Werkzeug werden?*

■ **Jamal:** Betrachtet man die unterschiedlichsten Anwendungsbereiche, in denen LabVIEW bereits eingesetzt wird, so lässt sich zweifelsohne von einer universell einsetzbaren Software-Plattform sprechen. Jede Entwicklung von LabVIEW wurde maßgeblich von den Anwendern getrieben. Auf diese Weise haben wir uns beispielsweise immer weiter in Richtung Automatisierungstechnik entwickelt. Wäre die Nachfrage bei LabVIEW nach objektorientierten Konzepten sehr groß gewesen, wären sie schon längst eingeflossen.

Rahman Jamal,
Technik- und Marketing-Chef bei National Instruments.

■ *Bei komplexen Automatisierungslösungen scheint der Bedarf nach einem einheitlichen Werkzeug für die Daten- und Managementebene aber sehr hoch zu sein ...*

■ **Jamal:** Die Praxis hat uns in den letzten Monaten gezeigt, dass diejenigen, die das Tool ObjectVIEW einsetzen, Anwender sind, die im objektorientierten Ansatz Kenntnisse besitzen. Sie hätten gern LabVIEW benutzt, vermissten bisher aber in diesem speziellen Bereich die Abbildung der textbasierten auf die grafische Welt. Insofern sehe ich das Zusatz-Tool als Bereicherung an, das LabVIEW einer noch breiteren Anwenderschicht zugänglich macht. LabVIEW wurde bewusst nicht an die objektorientierte Technologie angelehnt, sondern eher an das aufgabenorientierte Denken des Ingenieurs.

■ *Öffnen sich hierdurch nicht neue Anwendungsgebiete oder neue Kundenkreise?*

■ **Jamal:** Neue Anwendungsgebiete würde ich nicht sagen. Ich erwarte aber eine ganz neue Klasse von LabVIEW-Anwendern, die sich in der klassischen LabVIEW-Welt bisher nicht wiederfinden konnten. Nach meinem Ermessen spricht das Zusatz-Tool hauptsächlich die Anwender an, die bereits über Erfahrung mit objektorientierter Programmierung in einer textuellen Umgebung verfügen. Aber das wird der Markt und die Resonanz auf Anwenderforen zeigen.

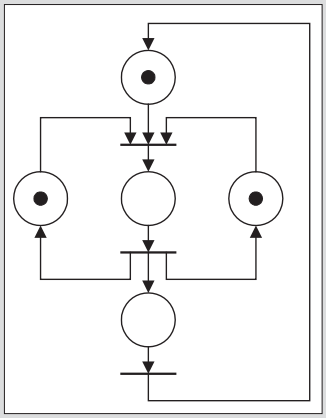
■ *Man hört immer mal wieder etwas von LabVIEW auf Chips. Wie weit ist denn dieser Gedankengang gediehen?*

■ **Jamal:** Seit einigen Jahren laufen Pilot-Projekte in unserem Haus, die sich mit dieser Thematik auseinandersetzen. Denkbar ist beispielsweise der Entwurf virtueller Instrumente mittels der grafischen Programmierumgebung LabVIEW. Die hiermit generierten Diagramme könnten dann als Funktionsbeschreibung für ein FPGA oder ASIC dienen. Erste Produkte können wir wahrscheinlich im Laufe des nächsten Jahres bekanntgeben. Was wir bereits dieses Jahr auf der MessComp vorgestellt haben, ist die Verteilung von LabVIEW-Programmen auf autarken I/O-Bausteinen – FieldPoint 2000. Es handelt sich sozusagen um die Vorstufe der LabVIEW-programmierbaren FPGAs.

Einen Haken hat die Sache aber. Wegen der Mulitasking-Fähigkeit von LabVIEW-Applikationen muss man aus mehreren Prozessen auf gleiche Parameter zugreifen können. Parametrierinformationen werden deshalb oft in „globale Variablen“ verwandelt. Das bringe Nachteile in puncto Sicherheit, erklärt Vogel. Es lasse sich nicht erkennen, welcher Prozess gerade zugreift. So könne es zu extremen Seiteneffekten kommen. Wie aber synchronisiert man den wechselseitigen Zugriff? Wenn überhaupt, wurde der Schutz bisher in LabVIEW per Hand, etwa über Semaphoren programmiert. Das sei sehr aufwendig und extrem unübersichtlich. ObjectVIEW regele nun den Zugriffsschutz automatisch. Während Teilnehmer A zugreift, existiert für weitere Teilnehmer eine Sperre. Besteht das

Geheimnis der Pertri-Netze

Ein Petri-Netz besteht aus drei Netzelementen (*Bild*): Plätze, Transitionen und Kanten. Übergänge, die zwischen den Plätzen liegen, bezeichnet man als Transitionen. In einem Petri-Netz sind Plätze und Transitionen durch gerichtete Kanten so verkettet, dass sie sich jeweils abwechseln. Zwischen zwei Plätzen befindet sich also immer ein Übergang. Im Petri-Netz wandern mindestens eine oder mehrere Marken. Ein Platz ist dann aktiv, wenn er eine Marke besitzt. Die Transitionen bestimmen, ob und wann ein Platz die Marke an seinen Nachfolger abgeben muss. Dafür besitzt jede Transition eine so genannte Weiterschaltbedingung. Ist diese erfüllt, erfolgt stets eine gerichtete Weitergabe der Marke, entsprechend den Kanten. Eine Marke kann sich während ihres Flusses in mehrere Marken aufteilen, die sich später vielleicht wieder zu einer Marke vereinen. Plätze und Transitionen werden verdrahtet. Die Verdrahtung symbolisiert die Kanten, über die in einem Petri-Netz die Marken fließen. In welchem Zustand sich Plätze befinden, ist abfragbar. Ein Virtuelles Instrument (VI) mit Petri-Netz kann sofort starten, vorausgesetzt, es ist fehlerfrei erstellt. Die Ereignisteuerung der Transitionen führt zu einer hohen Abarbeitungsgeschwindigkeit. Über Arrays lässt sich eine große Anzahl Vorgänger- und Nachfolgeplätze an jede Transition anschließen. Die Art der Implementierung von Weiterschaltbedingungen sichert einen konsistenten Zugriff auf die Kommunikationsobjekte. Folgende Petri-Netz-Typen sind möglich: Bedingungs-/Ereignisnetze (Plätze mit einer Marke), Stellen-/Transitionsnetze (Plätze mit mehreren Marken) und Prädikat-Transitionsnetze (Plätze tragen beliebige Daten). Petri-Netze sind ratsam wenn:



- der Objektlebenszyklus grafisch programmiert werden soll.
- ein System gleichzeitig mehrere Zustände annehmen kann.
- Zustände in mehrere Unterzustände unterteilbar sind.
- Zustände in einem System von vielen äußeren Bedingungen abhängig sind.
- Zustände in einem System teils abhängig und teils unabhängig voneinander sind.
- in der Programmstruktur eine nachgeschaltete Datenverarbeitung von der Zustandserkennung eines Prozesses getrennt sein soll.

SYSTEC
Gesellschaft für Automatisierung,
Systeme und Dokumentation mbH
Tel. 0911 – 99 89 55 0
Fax. 0911 – 99 89 55 633
Schreiberhauer Str. 5
90475 Nürnberg
www.systec-gmbh.com
info@systec-gmbh.com
Ein Unternehmen der Papp Gruppe
www.papp-gruppe.de

Applikationsprogramm also aus vielen globalen Variablen – in der Praxis machen das oft 50 Prozent aus –, sorgt das Werkzeug nun für erhöhte Sicherheit, bei weniger Programmieraufwand.

Petri-Netze schaffen Übersichtlichkeit

Eine gute Möglichkeit, komplexe parallele Abläufe und Prozesse beschreiben zu können, liegt in der Struktur der Petri-Netztechnik (siehe *Kasten* „Geheimnis der Petri-Netze"). Carl Adam Petri entwickelte bereits in den sechziger Jahren diesen genialen Ansatz. Das Jenaer Team hat ihn sich zu eigen gemacht und in ObjectVIEW integriert. Bei LabVIEW findet ein Datenaustausch statt. Mit den Zusatz-Tools kann jetzt auch der Fluss von Zuständen grafisch programmiert werden. Dies entpuppt sich als ein zusätzliches Abstraktionsniveau. Der Anwender besitzt erstmals die Möglichkeit, in einem Projekt beide Programmiermethoden beliebig zu kombinieren. Das Petri-Netz-Konzept schafft zum einen Übersichtlichkeit und Transparenz, zum anderen kann sich der Applikationsprogrammierer schrittweise einer detaillierten Problemlösung annähern. Häufig sind bei Projektbeginn die Strukturen von komplexen Automatisierungsaufgaben noch nicht genau bestimmbar. Mit der Petri-Netz-Philosophie lassen sich zunächst Grobstrukturen des Gesamtprozesses aufzeigen, etwa einer Ablaufsteuerung für Batch-Prozesse, ohne dass der Ablauf des Teilprozesses genau bekannt sein muss. Der Ingenieur setzt zunächst das Symbol für den Teilprozess ein und parametriert es. Die Realisierung des Teilprozesses und die Kommuni-

Interview ■ Denkbarrieren überwinden

Über strategische Hintergründe einer neuartigen grafischen Programmierumgebung für die Automatisierungstechnik sprach die *Elektronik* mit Jens Vogel, Geschäftsführer der Vogel Automatisierungstechnik GmbH.

Jens Vogel, Geschäftsführer der Vogel Automatisierungstechnik GmbH und Lehrbeauftragter für Objektorientierte Programmierung an der Fachhochschule Jena.

■ *Wie steht die alte Welt der Programmierer zu Ihrer neuen Entwicklung?*
■ **Vogel:** Die alteingesessenen C- oder Java-Programmierer glauben nicht, dass es möglich ist, mit grafischen Elementen einfach zu programmieren. Als Argumente höre ich häufig: „Das hat ja nichts mit richtigem Code zu tun. Das kann nicht funktionieren!"

■ *Bisher ist der Anwender Ihrer Tools an LabVIEW gebunden. Ist das nicht eine Einschränkung?*
■ **Vogel:** LabVIEW hat weltweit einen Marktanteil von über 60 Prozent und stellt sozusagen einen Quasistandard in diesem Bereich dar. Das impliziert doch ein interessantes Anwendungsvolumen.

● *LabVIEW ist durch die Ergänzung Ihrer Toolsets nun in der Lage, universelle Programmieraufgaben zu lösen. Ist National Instruments darüber begeistert?*
■ **Vogel:** Es wäre schön, wenn National Instruments im Hinblick auf neue Anwendungsgebiete noch einige Denkbarrieren überwinden könnte. Einer Symbiose würde dann nichts mehr im Weg stehen. Es ergibt sich für NI ein zusätzlicher großer Kundenkreis. Bisher unterstützt National Instruments den Vertrieb von ObjectVIEW nicht. Wir werden das Tool in unseren Automatisierungsprojekten einsetzen, aber auch einzeln anbieten.

■ *Können Sie in Zahlen ausdrücken, wie groß die Einsparung bei Projektierungs- und Wartungskosten durch ObjectVIEW ist?*
■ **Vogel:** Bei komplexen Programmen konnten wir bereits Einsparungen gegenüber konventioneller Programmierung von über 50 Prozent aufweisen. Das bestätigen uns auch die Anwender. Auch wenn kleine Bereiche einer Anlage zu einem späteren Zeitpunkt funktionstechnisch verändert oder aufgerüstet werden, lohnt sich der Einsatz: Die Modifikation liegt sofort grafisch dokumentiert vor und schafft hohe Transparenz.

■ *Schmiedet die Jetter AG nicht ähnliche Pläne mit ihrer Software-Umgebung?*
■ **Vogel:** Bisher existiert kein vergleichbares Tool auf dem Markt. Die Jetter AG agiert mit ihrer Software-Lösung hauptsächlich auf der Prozessebene. Hier gibt es auch die IDA-Standardisierungsbestrebungen für das Interface for Distributed Automation, die sich auf der SPS-Ebene abspielen. Der Funktionsplan soll auf mehrere Knoten verteilt werden, die dann Ereignisse austauschen. Vergleicht man IDA mit unserem Konzept, dann sind die Denkmuster identisch. Deshalb wird es auch einfach sein, beide Systeme zu koppeln. Auch der Entwurf nach IEC 61499, verteilte Steuerungssysteme zu schaffen, passt exakt zu unseren Objektnetzen. Wir nutzen die gleichen Funktionsprinzipien, nur eine Ebene höher. Wir sind genau das Pendant auf der Leitebene.

■ *Braucht man zukünftig eigentlich noch SPS-Programmierung? Sie könnten doch alles mit LabVIEW/ObjectVIEW erledigen.*
■ **Vogel:** Beide haben ihre Berechtigung. Die SPS-Geräte besitzen konstante Zykluszeiten, die sich berechnen lassen. Bei einer Ereignissteuerung sind die Reaktionszeiten abhängig von der Prozessorauslastung. Außerdem sind Petri-Netze viel universeller als die Programmiersprache der SPS. Wenn ein determinierter, begrenzter Leistungsumfang gefragt ist und Echtzeit sowie Sicherheitsaspekte im Vordergrund stehen, empfiehlt sich eine SPS. Bei beliebigen Leistungsmöglichkeiten und hohen Freiheitsanforderungen in Richtung PC- und IT-Welt werden LabVIEW und ObjectVIEW interessant.

■ *Wohin strebt der technologische Trend?*
■ **Vogel:** Agentensysteme liegen im Trend. Zurzeit werden sie in Java programmiert. Bei uns allerdings sind Objekte noch auf einzelnen Knoten beheimatet. Vorstellbar ist aber, dass in Zukunft in ObjectVIEW die Objekte ihren Knoten verlassen können und wandern. Sie erledigen dann Aufgaben auf unterschiedlichsten Plattformen. Danach kehren sie wieder zurück oder senden Informationen. Bisher sind das Visionen, die aber aufgrund der Plattformunabhängigkeit technisch möglich sind.

kation übernimmt dann ein aktives Objekt mit einem selbstständigen Prozess.

Es existieren verschiedene Petri-Netz-Typen. Mehrere Petri-Netze lassen sich hierarchisch strukturieren und synchronisieren. Ebenso können Petri-Netze in aktiven Objekten untergebracht sein. Die Beschreibung des Verhaltens eines Objekts erfolgt dann mit Hilfe eines Petri-Netzes. Umgekehrt lassen sich aber auch Objekte auf den Plätzen von Petri-Netzen (Referenznetze) positionieren. Ein aktives Objekt, das wiederum aktive Objekte enthält, wird als Objektnetz bezeichnet. Ein Objektnetz, kombiniert mit Petri-Netz-Technik, könnte im Beispiel „Verladestelle mit Zementsilos“ so aussehen (*Bild 6*): Eine Verladestelle besteht aus einer Waage und drei Silos. Für jedes der drei Silos wird jeweils eine Instanz eines Silo-Objekts erzeugt. Das Silo als aktives Objekt reagiert ereignisgesteuert und unabhängig von der Verladestelle auf Änderungen an seinen Ports. Ein Petri-Netz repräsentiert das Verhalten. Das Silo-Objekt kommuniziert selbstständig mit der Hardware, der Applikationsschicht und dem User-Interface. Im übergeordneten Objektnetz „Verladestelle“ der Applikationsschicht ist die Verschaltung der Silos und der weiteren Komponenten (aktiven Objekte) der Verladestelle realisiert. Der Vorteil dieses Objektnetzes liegt hauptsächlich darin, dass auf verschiedenen Abstraktionsebenen gearbeitet und somit die Applikation automatisch hierarchisch strukturiert wird. Die Implementierung ist von der Struktur her mit dem Entwurf identisch. Das heißt, das Programm dokumentiert sich auch in seiner Struktur selbst. Jedes Silo agiert unabhängig und ereignisgesteuert. Seine Funktion wird eine Ebene tiefer festgelegt und gekapselt. Man könnte also jetzt die Funktion des Silos ändern, ohne dabei das übergeordnete Netz – die Verladestelle – zu beeinträchtigen. Hier wird das Zusammenspiel der Silos und der Waage beschrieben. Umgekehrt lassen sich Silos hinzufügen, ohne dass man wissen muss, wie das einzelne arbeitet. Das bringe, so die Entwickler, enorme Kosten- und Zeitvorteile, sowohl bei der Programmierung als auch bei der Wartung von Anlagen (siehe *Kasten* „Denkbarrieren überwinden“) oder falls später erweitert oder modifiziert werden müsse. Da in der Industrie Mess- und Automatisierungslösungen oft jahrzehntelang im Einsatz sind, werden verteilte intelligente Software-Strukturen, die auf einfache Weise grafisch programmierbar sind, umso dringlicher. Selbst bei kleineren Projekten lohne sich der Einsatz von ObjectVIEW schon, betont Vogel.

Das Petri-Netz lässt sich direkt im Blockdiagramm von LabVIEW erstellen und editieren. Die Symbolik entspricht der allgemein üblichen Notation für Petri-Netze, wobei die grafischen Elemente von LabVIEW derzeit noch kleine Kompromisse nötig machen. Es fehlt etwa die Möglichkeit, runde Ecken zu gestalten oder schräge Linien zu zeichnen. Auch die Farbe ist nicht frei wählbar. National Instruments hat allerdings mit der nächsten LabVIEW-Version angekündigt, dies zu ermöglichen.

SYSTEC
Gesellschaft für Automatisierung,
Systeme und Dokumentation mbH
Tel. 0911 – 99 89 55 0
Fax. 0911 – 99 89 55 633
Schreiberhauer Str. 5
90475 Nürnberg
www.systec-gmbh.com
info@systec-gmbh.com
Ein Unternehmen der Papp Gruppe
www.papp-gruppe.de

Chance zu neuen Anwendungsgebieten

Die Einsatzmöglichkeiten, die sich durch die Erweiterung von LabVIEW zu ObjectVIEW eröffnen, schätzt Vogel als gewaltig ein. Prinzipiell seien sehr viele softwaretechnische Probleme damit lösbar. In den letzten Monaten hielt die Plattform denn auch Einzug in die Praxis. In einem Zementwerk wurde der Versand automatisiert. Die Firma Systec entwickelt zur Zeit ein völlig neues Mess- und Automatisierungskonzept für Prüfstände, das auf ObjectVIEW basieren wird. Das „Universal Test and Automation System (UTAS)“ besteht aus mehreren netzwerkbasierten Komponenten (Motor, SPS, PC, ...), die ihre Aufgaben teils abhängig, teils unabhängig voneinander erfüllen müssen. „Diese Struktur lässt sich nun durch ereignisgesteuerte, aktive Objekte abbilden“, so Herbert Pichlik, Manager Testsysteme der Systec GmbH. Das sei mit LabVIEW bisher nicht so einfach möglich gewesen. LabVIEW-Entwickler seien jetzt sogar in der Lage, in völlig neue Terrains der Informationstechnologie vorzustoßen, etwa in den Bereich des Workflow-Managements, meint Jens Vogel. Nach Vogels Meinung erhalte LabVIEW die Chance, zum universellen Programmierwerkzeug aufzusteigen und neue Anwendungsgebiete zu erobern.

Neue Denkmuster erfordern neue Methoden

Die Einführung und die Anwendung objektorientierter Methoden ist ohne Zweifel anfangs aufwendig. Die Arbeitsabläufe ändern sich, anspruchsvolle Weiterbildung der Mitarbeiter ist erforderlich und neue Werkzeuge werden benötigt. Aber:

- Man revolutioniert den gesamten Prozess vom Systementwurf bis zur Wartung.
- Man erhält wiederverwertbare und zuverlässige Software-Komponenten.
- Man erhöht die Sicherheit der Integration von Objekten.

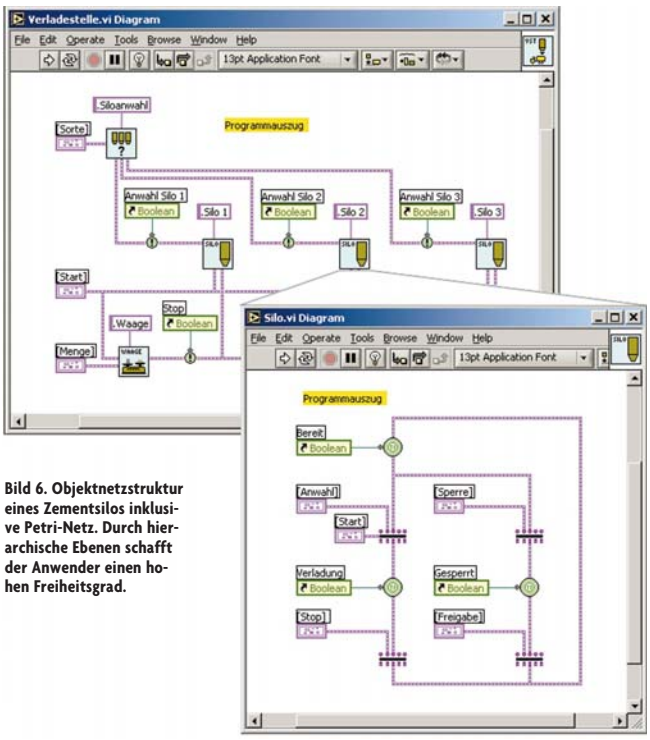


Bild 6. Objektnetzstruktur eines Zementsilos inklusive Petri-Netz. Durch hierarchische Ebenen schafft der Anwender einen hohen Freiheitsgrad.

Herbert Pichlik
ist Manager Testsysteme bei der Systec GmbH. Nebenberuflich ist er Lehrbeauftragter für LabVIEW und Virtuelle Instrumentierung an der Georg-Simon-Ohm-Fachhochschule in Nürnberg.
E-Mail: hpichlik@systec-gmbh.com

Hilde-Josephine Post
ist freie Journalistin in München.

- Man verfügt über ein Reservoir von modularen Funktionseinheiten, die sich zu neuen, übersichtlichen Systemen kombinieren lassen.
- Man gewinnt einen zu 100 % EDV-gestützten, formalisierten Lebenszyklus der Systeme.
- Man erzielt einen lückenlosen Übergang von der prüfbaren Modellierung bis zur Software-Implementation.
- Man gewinnt die Wiederholbarkeit des Durchlaufens von vollständigen Entwicklungsrhythmen nach jeder Änderung, Ergänzung und Erweiterung.
- Man archiviert in verständlicher, dokumentierter und direkt wiederverwendbarer, hardware-unabhängiger Form Know-how auf Detailgebieten und Spezialwissen von Mitarbeitern.

ObjectVIEW bietet erstmals die Möglichkeit, die Vorteile der Objekt-Technologie mit der intuitiven Arbeitsweise der grafischen Programmierung zu verbinden. Der Einstieg in die neue Denkweise kann durch Schulungen und / oder Einstiegs-Applikationen durch Vogel Automatisierungstechnik GmbH wesentlich verkürzt werden. *jk*

Weitere Informationen unter:

www.ObjectVIEW.de
www.gooptech.com
www.vat.de
www.systec-gmbh.com
www.ni.com