

Experten-Interview zu Software-Suite und Entwicklungsumgebung

## Die Funktionalität ist maßgebend

**Die Reduzierung des Engineeringaufwands ist bei den immer komplexeren Automatisierungsstrukturen schon lange ein Topthema. Trumpf ist hier die möglichst optimale Durchgängigkeit von Entwicklungsumgebungen bzw. Software-Suiten. Doch was ist unter diesen in vielfältigster Weise verwendeten Begriffen eigentlich genau zu verstehen und welche unterschiedlichen Anwendungsvorteile sind damit verbunden? Diese Frage beantworten im Rahmen des Trendteils „Integrated Manufacturing“ die Experten von 3S-Smart Software Solutions, B&R, Beckhoff, CopaData, Elau, Mitsubishi Electric, National Instruments, Phoenix Contact, Siemens, Sigmatek und Wonderware.**

**elektro AUTOMATION:** *Wie ist aus Ihrer Sicht eine Entwicklungsumgebung und wie eine Software-Suite zu definieren? Gibt es hier überhaupt einen Unterschied?*

**Gangl:** In diesem Bereich herrscht ein Wirrwarr bei der Begriffsdefinition. Egal ob man von Entwicklungsumgebung oder Suite spricht: Für den Anwender steht die Einfachheit der Handhabung und das all-in-one der Software im Vordergrund. Die Software-Suite Lasal von Sigmatek umfasst Tools zur objektorientierten Programmierung, Visualisierung, für Motion Control, zur Parametrierung der Sicherheitslogik so-



Bernhard Gangl von Sigmatek

wie Diagnose und Fernwartung. Somit steht dem Anwender eine durchgängige Entwicklungsplattform zur Verfügung, mit der sich die gesamte Applikation schnell und flexibel realisieren lässt.

**Hägele:** Eine Software-Suite ist per Definition die Zusammenfassung verschiedener Softwareanwendungen zu einem Gesamtpaket. Als Softwareanbieter vertritt CopaData diesen Ansatz und bietet seinen Kun-

den und Interessenten unter dem Namen Zenon eine modular aufgebaute Suite an. Je nach Aufgabenstellung ist diese durchgängig skalierbar von der Funktionstiefe eines einfachen HMI bis hin zu überlagerten Anwendungen in den Bereichen Scada, Produktionsleittechnik und MES. Der Anwender entscheidet bedarfsgerecht, welche Funktionen er nutzen möchte, um eine für seine Bedürfnisse maßgeschneiderte Lösung zu entwickeln und umzusetzen. Wichtig dabei ist, dass unsere Suite nicht als gekapselte Lösung zu verstehen ist. Unsere Software Zenon gewährleistet Interoperabilität und ermöglicht es, Fremdsysteme einfach und effizient anzubinden – seien es CAD-Programme, Auftragsysteme, Engineering-Tools, Übersetzungs- und Dokumentationswerkzeuge, Visual Studio (VSTA) oder verschiedenste SPS-Systeme über Direkttreiber oder OPC UA.

**Hartkopf:** Wir sehen einen deutlichen Unterschied zwischen den beiden Begrifflichkeiten. Unsere Entwicklungsumgebungen sind Anwendungsprogramme, die es ermöglichen, unsere Hardware zu programmieren, Programme zu debuggen oder zu simulieren. Eine Software-Suite hingegen enthält mehrere Entwicklungsumgebungen und ermöglicht den einfachen Datenaustausch und eine Querkommunikation zwischen den unterschiedlichen Umgebungen. Zur Verdeutlichung: Die demnächst releaste Software Suite iQ Works von Mitsubishi Electric integriert folgende Entwicklungsumgebungen: GX Works für SPS-Programmierung, GT Works für GOT-HMI-Programmierung, MT Works für Motion-Controller-Projekte.

**Hess:** Hier gibt es keine genauen Begriffsdefinitionen. Aber der Name Software-Suite zielt ja darauf ab, dass in einer einzelnen



Dieter Hess von 3S-Smart Software Solutions

Software verschiedene Aspekte des Engineerings abgedeckt sind, nicht nur die reine Entwicklung von Applikationssoftware. Das ist natürlich essenziell für die Automatisierungstechnik, bei der die Bereiche Programmierung, Feldbuskonfiguration, HMI/Scada, Antriebs-Konfiguration, Bewegungsplanung und Konfiguration der Sicherheitstechnik vorkommen. Ferner ist eine Integration von Elektro-CAD, Simulation sowie Dokumentation der Maschinenbedienung und der kaufmännischen Software (Angebotswesen, Materialbeschaffung) denkbar. Zwar könnten die letztgenannten Aspekte ebenfalls in einer Suite enthalten sein, da aber bereits leistungsfähige Software dafür am Markt eingeführt ist, ist hier die Schaffung von Schnittstellen das Mittel der Wahl.

**Jamal:** Beschränkt man sich auf unsere Branche, spricht die Mess- und Automatisierungstechnik, so lassen sich klare Unter-



Rahman Jamal von  
National Instruments

schiede zwischen einer Software-Suite und einer Entwicklungsumgebung feststellen. Eine Software-Suite ist im allgemeinen Fall eine reine Sammlung von losen Bibliotheken, die oft eine mehr oder weniger gemeinsame Benutzeroberfläche nutzen. Bei höherwertigen Software-Suites lassen sich die Daten zwischen den Bibliotheken austauschen. Integraler Bestandteil einer Entwicklungsumgebung wie NI Labview hingegen ist eine – in unserem Beispiel grafische – Programmiersprache gekoppelt mit einem Compiler, Debugger sowie mit Werkzeugen für die Quellcode- und Projektverwaltung, Diagnosewerkzeugen und der automatischen Erstellung von Benutzeroberflächen während der Applikationsprogrammierung. Hiermit lässt sich eine Idee bis zur ausführbaren Applikation eigenständig entwicklungsnahe testen und umsetzen. In Bezug auf die Integration von anderen Werkzeugen, beispielsweise C-Dialekten, .Net, Design- und Mathematik-Werkzeugen, ist eine Entwicklungsumgebung wie Labview völlig offen.

**Papenfort:** Eine Entwicklungsumgebung ist normalerweise das Werkzeug, um mit einer Programmiersprache Lösungen zu erstellen. Natürlich ist in so einem Werkzeug all das integriert, was man zum komfortablen Arbeiten braucht. Das sind neben einem guten Debugger auch Quellcodeverwaltungsdatenbanken. In eine Software-Suite sind zusätzlich mehrere Programmiersprachen, Konfiguratoren und Diagnosewerkzeuge integriert. Die neue Softwaregeneration TwinCAT 3 kann als eine Software-Suite aufgefasst werden. Hier befinden sich unter einem Dach – dem Microsoft Visual Studio –



Dr. Josef Papenfort  
von Beckhoff

unterschiedliche Werkzeuge, wie Editoren für IEC 61131, C/C++ oder C#; daneben natürlich noch Konfiguratoren wie der TwinCAT System Manager für Feldbusse und Motion sowie weitere Werkzeuge.

**Paulin:** Eine Software-Suite ist aus unserer Sicht eine Sammlung von Programmen, die nur lose miteinander verbunden sind. Eine Entwicklungsumgebung hingegen definiert sich bei B&R durch das Ziel, den Anwender bei der Umsetzung einer Idee vollumfänglich zu unterstützen. Alle notwendigen Werkzeuge sind darin integriert und arbeiten auf einer gemeinsamen Basis eng verzahnt zusammen. B&R bietet mit dem Automation Studio seit Jahren ein Werkzeug zur integrierten Automatisierung von Maschinen und Anlagen. Mit einem Werkzeug werden Steuerungen im Team programmiert, mit Sicherheitstechnik ausgestattet, Visualisierungen erstellt und Antriebe projektiert. Umfassende Diagnosemöglichkeiten runden dieses Werkzeug ab.

**Wegener:** Unter dem Begriff Entwicklungsumgebung verstehen wir eine strikt abgegrenzte Software-Umgebung oder Tools für die Entwicklung eigener SPS-Applikationen.

Dabei sind alle Funktionen gegeben, die der Anwender für eine schnelle und fehlerfreie Erstellung einer Anwendung benötigt. Dazu gehören Debug- und andere Testfunktionen. Der Begriff Software-Suite ist hier wesentlich weiter gefasst. Die Software-Suite umfasst unterschiedliche Werkzeuge für die verschiedenen Phasen der Wertschöpfungskette, angefangen bei Planungswerkzeu-



Friedrich Wegener von  
Phoenix Contact

gen, die den Planer bei der Auswahl der notwendigen Geräte unterstützt, bis hin zu Diagnose- und Wartungswerkzeugen in der Betriebsphase. Anwender dieser Werkzeuge sind in der Regel unterschiedliche Personen, mit spezifischem Fachwissen für das jeweilige Anwendungsgebiet der Werkzeuge. ▶

## DIE EXPERTEN

- **Bernhard Gangl**, Leiter Software bei der **Sigmathek GmbH & Co KG** in **Lamprechtshausen/A** ([www.sigmatek-automation.com](http://www.sigmatek-automation.com))
- **Frank Hägele**, Sales Manager der **Copa-Data GmbH** in **Ottobrunn** ([www.copadata.de](http://www.copadata.de))
- **André Hartkopf**, Product Manager Software bei **Mitsubishi Electric B.V., Factory Automation – European Business Group**, in **Ratingen** ([www.mitsubishi-automation.de](http://www.mitsubishi-automation.de))
- **Dieter Hess**, Geschäftsführer der **3S-Software Solutions GmbH** in **Kempten** ([www.3s-software.com](http://www.3s-software.com))
- **Rahman Jamal**, Technical & Marketing Director Central Europe der **National Instruments Germany GmbH** in **München** ([www.ni.com/germany](http://www.ni.com/germany))
- **Dr. Josef Papenfort**, Produktmanager TwinCAT bei der **Beckhoff Automation GmbH** in **Verl** ([www.beckhoff.com](http://www.beckhoff.com))
- **Werner Paulin**, System Architect bei der **Bernecker + Rainer Industrie-Elektronik Ges.m.b.H. (B&R)** in **Eggelsberg/A** ([www.br-automation.com](http://www.br-automation.com))
- **Friedrich Wegener**, Mitarbeiter des Systemmarketings in der **Business Unit Automation** bei der **Phoenix Contact Electronics GmbH** in **Bad Pyrmont** ([www.phoenixcontact.de](http://www.phoenixcontact.de))
- **Thorsten Weiler**, Regional Sales Manager der **Wonderware GmbH** in **München** ([www.wonderware.de](http://www.wonderware.de))
- **Stefan Werner**, Produktmanager Simatic Engineering Software der **Siemens-Division Industry Automation** in **Nürnberg** ([www.siemens.com/automation](http://www.siemens.com/automation))
- **Klaus Weyer**, Leiter strategisches Marketing bei der **Elau GmbH** in **Marktheidenfeld** ([www.elau.de](http://www.elau.de))

**Weiler:** Diese Begriffe sind bei Wonderware deutlich voneinander getrennt. Eine Software-Suite ist ein Paket bestehend aus einzelnen Software-Komponenten, die unabhängig voneinander einsetzbar sind. Klassisch ist hier der Vergleich zum Microsoft-Office-Paket mit den Komponenten Word, Excel, Powerpoint etc. In der Automatisierungssoftware hat jede Komponente einer Suite eine eigene Entwicklungsumgebung. Mit der Entwicklung der Wonderware System Platform als durchgängige Automatisierungssoftware-Plattform haben wir diesen Ansatz zu einer zentralen, einheitlichen und integrierten Entwicklungsumgebung ohne Schnittstellen weiterentwickelt.

**Werner:** Eine Software-Suite ist lediglich eine lose Kopplung verschiedener eigenständiger Tools. In einer Entwicklungsumgebung werden hingegen alle benötigten Funktionalitäten zur Applikationserstellung zusammengeführt. Die Verwaltung von Querabhängigkeiten in einem Projekt wird – im Gegensatz zur Software-Suite – von der Entwicklungsumgebung geleistet.

**Weyer:** Bevor ich auf Ihre Frage antworte, müssten Sie definieren, welche Leistungsmerkmale Sie welchem Begriff zuordnen. Soweit ich für Elau sprechen kann, wir sehen hier keinen Unterschied. Beide Begriffe stehen für ein Tool-Konzept, welches Software-Werkzeuge zur Entwicklung einer Lösung umfasst. Das kann man sehr gut auch am Beispiel PacDrive demonstrieren: Für PacDrive M haben wir Tools zum Programmieren der Applikation und der HMI-Lösung, Tools für Bewegungsdesign und Daten-Handling sowie zur Diagnose. Die gleichen Tools nutzen wir in weiterentwickelter Form wieder für SoMachine Motion, unserer Workbench für PacDrive 3. Natürlich



Klaus Weyer  
von Elau

bietet SoMachine Motion gegenüber dem alten Tool-Konzept einen erweiterten Funktionsumfang, der zukünftig noch beträchtlich wachsen wird. Aber ob Tool-Konzept, Workbench oder auch Software-Suite, keiner dieser Begriffe differenziert sich vom anderen per se durch einen größeren oder kleineren Funktionsumfang.

**elektro AUTOMATION:** Welche Vor- und Nachteile ergeben die wichtigsten Unterschiede ganz konkret für den Anwender?

**Gangl:** Verschiedene Aufgaben erfordern unterschiedliche Benutzeroberflächen. Eine Entwicklungsumgebung für Programmierung und Debugging hat andere Anforderungen als sie für das Erstellen von Bildschirm-Eingabemasken notwendig sind. Diverse Zeichenfunktionalitäten haben in einer Programmierumgebung nichts zu suchen. Wichtig sind aber eine einheitliche Bedienphilosophie und die Möglichkeit, Daten elegant zwischen den verschiedenen Benutzeroberflächen auszutauschen sowie Projekt-elemente nur an einer zentralen Stelle verwalten zu müssen.

**Hägele:** Wir haben den Ansatz einer modular aufgebauten, integrierten Software-Suite gewählt. Die Durchgängigkeit von HMI bis MES bietet den Anwendern signifikante Einsparungsmöglichkeiten. Er investiert anstatt in zwei oder mehr Systeme nur noch in eine Software, um unterschiedlichste Aufgabenstellungen zu lösen. In der Applikationserstellung und -pflege können durch die konsistente Projekt- und Datenbasis die Engineeringzeiten signifikant verkürzt werden. Der Softwareentwickler oder Instandhalter muss darüber hinaus nur noch auf ein System geschult werden. Jedoch sollte ein Anwender stets darauf achten, ein offenes und herstellerunabhängiges System wie zenon einzusetzen. Klassischerweise reduzieren sich Anbieter von Software-Suiten auf deren eigenes Produktportfolio in punkto Software und Hardware. Damit begibt sich der Anwender langfristig in eine starke Abhängigkeit und kann nicht auf innovative Technologien anderer Anbieter reagieren.

**Hartkopf:** Hervorzuheben ist der einfache Datenaustausch zwischen den Einzelpaketen einer Suite. Diese Durchgängigkeit verringert deutlich Fehlerquellen. Ein weiterer Punkt ist das bedienerfreundliche Handling. Mit geringem Trainingsaufwand lässt sich Basiswissen für das Gesamtpaket vermitteln und es bietet dem Anwender die Möglichkeit, zügig mit der Programmierung zu starten. Auch dadurch ergeben sich reduzierte Engineeringzeiten. Die Projektierungskosten insgesamt werden durch die Integration der Pakete in einer Suite gesenkt. Nachteil: Da in einer Suite teilweise viele Tools und Programmierwerkzeuge für die unterschiedlichsten Anwendungszwecke vereint werden, kann diese in Struktur und Handhabung sehr komplex werden. Ein großer Vorteil der Entwicklungsumgebungen ist, dass sie individuell und optimal für den jeweiligen Einsatzzweck konzipiert sind. Ein

Nachteil liegt darin, dass sie nicht immer System übergreifend, sondern nur auf die jeweilige Hardware zugeschnitten sind.

**Hess:** Ob der Anwender Vorteile hat, hängt natürlich von der Verzahnung der Teile einer Suite ab. Da gibt es große Unterschiede. Im einfachsten Fall ist nur die Oberfläche integriert: Es gibt einen gemeinsamen Navigations-Baum und ein gemeinsames Rahmenfenster. Fortgeschrittene Software-Suiten erlauben den Austausch von Projektierungsdaten auf dem kleinsten gemeinsamen Nenner, typischerweise Datenpunkte/Symbolisten. Der größte Vorteil ergibt sich für den Anwender, wenn die einzelnen Engineering-Aspekte das gleiche Programmier- und Konfigurationsparadigma haben. In diesem Fall sind die Konstrukte in allen Teilen der Suite dieselben. Beispielsweise kennt die Visualisierung oder die Feldbuskonfiguration dann einen Funktionsbaustein oder sogar eine Klasse und deren Eigenschaften, oder ein Bibliothekskonzept steht allen Teilen in gleicher Weise zur Verfügung – um nur zwei Beispiele zu nennen. Leider entsteht so eine Software-Suite nie aus der Integration bestehender Software, sondern nur durch eine komplette Neuentwicklung. Daher sind solche Systeme bisher nicht allzu häufig.

**Jamal:** Der größte Vorteil einer Entwicklungsumgebung liegt klar auf der Hand: In der Entwicklungsumgebung steht dem Anwender alles zur Verfügung, was er zur Applikationserstellung braucht. Die zeit- und kostenintensive Umstellung auf immer wieder neue Werkzeuge ist hinfällig. Die durchgängige Methodik der grafischen Programmierung wie bei Labview ermöglicht die interdisziplinäre Erstellung von Anwendungen auch in Teams. Das Spektrum der Applikationen reicht dabei von Design über Entwicklung bis hin zur Serienfertigung und deckt die unterschiedlichsten Domänen ab, wie Mess- und Prüftechnik, industrielle Steuerung und Regelung, anspruchsvolle Mathematik, Bildverarbeitung, Motorsteuerung bis hin zum Embedded Design.

**Papenfort:** Der große Vorteil einer Software-Suite ist der Gedanke, alles unter einem Dach zu vereinigen. Damit wird die Einarbeitungszeit reduziert, da die Bedienung vereinheitlicht ist. Bei Twincat 3 ist das Dach Microsoft Visual Studio, das weltweit eines der bekanntesten Werkzeuge zur Erstellung von C/C++ oder .Net-Applikationen ist. Der Anwender profitiert natürlich auch von den vorhandenen Werkzeugen, die sich in Visual Studio integrieren. Dazu gehören z.B. bekannte Quellcodeverwaltungstools wie Subversion oder der Microsoft Team Foundation Server.



Werner Paulin  
von B&R

**Paulin:** Eine Entwicklungsumgebung wie das Automation Studio bietet dem Anwender einen einheitlichen, zentralen Zugang für die Umsetzung einer Automatisierungslösung. Durch die nahtlose Integration und die gemeinsamen Datenbasis aller Komponenten entfallen zeitaufwändige Abstimmungen. Der Anwender kommt dadurch schneller zum Ziel. Die einheitliche Bedienphilosophie reduziert zusätzlich den notwendigen Schulungsaufwand. Mit der Möglichkeit, in einem Projekt beliebig viele Maschinenvarianten mit einer gemeinsamen Softwarebasis zu verwalten, sichert der Einsatz des Automation Studios nachhaltig Investitionskosten in der Entwicklung von Maschinen-Software.

**Wegener:** Entscheidend bei dem Suite-Gedanken sind das optimale Zusammenspiel der unterschiedlichen Werkzeuge für alle Phasen und der damit einhergehende Datenfluss von einem vorgelagerten in das jeweils nachgelagerte Werkzeug und ggf. umgekehrt. Wesentlicher Bestandteil einer Suite wie der Automationworx Software Suite ist in der Regel auch eine Entwicklungsumgebung – hier PCWorx – für das verwendete Steuerungssystem. Die Nutzung einer in eine Software-Suite eingebetteten Entwicklungsumgebung bringt wesentliche Vorteile hinsichtlich der Datendurchgängigkeit. So können Daten für die Netzwerkkonfiguration aus der Planungsphase direkt übernommen oder aber Programmvariablen und Programmbausteine direkt in der nachgelagerten Visualisierung für eine teils sogar automatische Visualisierungserstellung genutzt werden. Während der Entwicklungsphase entstandene Informationen können wieder in der Betriebsphase für Diagnosezwecke genutzt werden. So entfallen redundante Eingaben.

**Weiler:** Der Vorteil einer Suite ist die Möglichkeit einer voneinander unabhängigen Nutzung der Komponenten. Weniger praktisch bei einer Suite: Schnittstellenpflege, das Fehlen eines einheitlichen Datenmodells und die Bearbeitung einer Applikation mit unterschiedlichen Programmen. Vorteile einer einheitlichen Entwicklungsumgebung –

wie der Wonderware IDE – inkl. zentraler, intuitiver Oberfläche, in der das Datenmodell und Projekt verwaltet wird – sind hingegen der deutlich geringere Aufwand z.B. für Schulung der Anwender und Administration der Applikation, Zeit- und Kostenersparnis durch eine zentralisierte Entwicklung sowie eine verbesserte Zusammenarbeit. So können mehrere Entwickler gleichzeitig an einer Applikation arbeiten und Eingaben der Kollegen sofort berücksichtigen. Änderungen der Applikation werden automatisch über das Netzwerk auf die jeweiligen Client-Rechner verteilt.

**Werner:** Eine Entwicklungsumgebung ist für die Erstellung von bestimmten Applikationen optimiert. Durch die Integration verschiedener Funktionalitäten können Daten meist einfach übernommen und innerhalb der Entwicklungsumgebung konsistent ge-



Stefan Werner von  
Siemens

halten werden. Das Arbeiten mit einer Entwicklungsumgebung wird zusätzlich durch eine durchgängige Usability vereinfacht. Eine Software-Suite bietet diese Vorteile nicht. Jede Applikation hat ihren eigenen Datenhaushalt, was für den Anwender zeitaufwändige und fehlerträchtige Doppelergaben zur Folge hat.

**Weyer:** Der Begriff der Software-Suite dient gerne als Marketing-Instrument dazu, dem Anwender die Vorstellung eines leistungsfähigen, lückenlosen Tool-Konzepts zu projizieren. In der Praxis sieht das nach meinem Eindruck dann so aus, dass die einzelnen Tools nicht mehr als Icons auf dem Desktop des PCs liegen, sondern von einer herstellereigenen Oberfläche aus gestartet werden. Das ist mit Blick auf die Übersichtlichkeit ja auch ein Schritt in die richtige Richtung, das zeigt auch SoMachine Motion: Sie haben eine CD, sie führen einen Installationsprozess durch. Dann sind alle Tools und die freien Bibliotheken auf einer Oberfläche zu finden, darüber hinaus eine komplette Dokumentation der Pac-Drive-3-Hardware. Aber wie bereits gesagt, das ist kein echtes Differenzierungsmerkmal zwischen den Begriffen Entwicklungsumgebung und Software-Suite. ▶

**elektro AUTOMATION:** Welche mittelfristigen Entwicklungs- bzw. Leistungstrends erwarten Sie im Bereich dieser Automatisierungssoftware?

**Gangl:** Die Programmierweise entwickelte sich bis dato im Maschinenbau von Assembler oder grafischen Programmiersprachen wie KOP hin zur „einfachen“ Hochsprache Structured Text. Es gibt zunehmend mehr Entwickler, die durch ihre Ausbildung mit mächtigen, ressourcenhungrigen Programmiersprachen unter Windows, Linux etc. vertraut sind, und nun in die Steuerungsprogrammierung mit harten Echtzeitanforderungen eintauchen. Um diese beiden Programmierwelten zu verbinden, eignet sich die einfach anzuwendende Objektorientierung, auf die Sigmatek bereits seit 2000 setzt, optimal. Hinzukommen werden Anforderungen wie UML Tools. Die Aufgabe wird weiterhin sein, den studierten Hochsprachenprogrammierer gleichermaßen zufrieden zu stellen wie den konventionellen Steuerungsprogrammierer.

**Hägele:** Die beiden bedeutendsten Trends in der Automatisierungssoftware sind sicher Offenheit und vertikale Integration. Eine offene Lösung muss Treibervielfalt bieten und konsequent alle marktgängigen Plattformen, Standards sowie Anwendungen unterstützen. Das schafft Unabhängigkeit für den Kunden in der Wahl der Hard- und Software, der Partner und für die spätere Erweiterung. Gleichzeitig steigt damit auch die Produktivität der Anwender – Aufwand und Kosten sinken. Mit über 300 Kommunikationsprotokollen zu allen gängigen Steuerungen, unseren Zertifizierungen als ‚Microsoft Gold Certified Partner‘ und als ‚SAP Software Solutions Partner‘ sowie unserem zertifizierten, bidirektionalen SAP-Interface belegen wir, dass wir diese Anforderungen



Frank Hägele von  
Copa-Data

ernst nehmen. Dies findet auch hohe Akzeptanz in den Unternehmen, denn für viele unserer Kunden und Interessenten steht die vertikale Integration ihrer IT-Lösungen heute und in naher Zukunft im Fokus – von der ERP-Lösung über die MES-Anwendung bis hin zum Fertigungssystem.



André Hartkopf von  
Mitsubishi Electric

**Hartkopf:** Der Trend geht vermehrt in Richtung objektorientierte Programmierung. Ein einheitliches Datenformat („common exchange format“) ist ein Lösungsansatz, um mit Fremdanbietern Daten auszutauschen und damit die Lieferanten übergreifende Durchgängigkeit zu verbessern. Suites können mit ihrem durchgängigen Informationshandling eine Lösung vom Sensor bis zum ERP-System abbilden und helfen dadurch, Produktionsabläufe von der Produktion bis zur Managementebene transparenter zu machen.

**Hess:** Ich sehe neben der Steigerung des vorher erwähnten Integrationsgrads drei Trends: erstens die Erweiterbarkeit einer Automatisierungssoftware durch neue Module, typischerweise durch den Hersteller, aber in Zukunft durchaus durch dritte Firmen; zweitens die Integration von bekannten Tools aus der klassischen IT-Entwicklungswelt, wie Quellcodeverwaltung, Entwurfswerkzeuge (UML), Tools zur Testgenerierung/-automatisierung, statische Codeanalyse (Lint) und Profiler; drittens die bessere Unterstützung typischer Entwicklungsvorgehen für die Automatisierungstechnik, idealerweise sogar kundenspezifisch anpassbar. Durch Letzteres zerfällt jede Entwicklung einer Applikation in die Technologie-Entwicklung, die Auswahl und Parametrierung dieser Technologie-Funktionen, die Adaption an konkrete Installation (z.B. E/A-Zuordnung) und die Inbetriebnahme. Heutige Software tut meist so, als würde jede Applikation mit allen Aspekten von Grund auf neu entwickelt. Alle Teilaufgaben außer der Technologie-Entwicklung sind relativ schematisch und bieten durch bessere Toolunterstützung ein hohes Rationalisierungspotenzial.

**Jamal:** Der Trend „die Software bestimmt die Funktionalität der Hardware“ wird sich mittel- und langfristig noch rasanter fortsetzen. Hierbei nimmt eine Entwicklungsumgebung eine Schlüsselrolle ein, die von einer Software-Suite kaum getragen werden kann. Die nahtlose Abstimmung zwischen Software und Hardware inklusive hochperformanter Multicores und FPGAs wie es bei National Instruments der Fall ist, wird sich weiterhin auf die Leistungsfähigkeit und

Kosteneffizienz von Systemen sowie die Produktivität von Anwendern niederschlagen und somit insgesamt deren Wettbewerbsfähigkeit erhöhen.

**Papenfort:** Mittel- und langfristig wird der Anteil der Software in der Automatisierungstechnik enorm steigen. PC-basierte Steuerungstechnik mit immer leistungsfähigeren PCs bietet hier die ideale Plattform. Mehr Software bedeutet aber auch mehr Engineeringkosten für diese Software. Um die Kosten langfristig im Griff zu halten, ist es nötig, mit entsprechenden Tools die Softwareerstellung zu vereinfachen. Hierzu zählt auch objektorientiertes Programmieren – wie in der 3rd Edition der IEC 61131-3 beschrieben – und eine modulare Softwareerstellung. Modulare, einfach wiederzuwendende Software ist der Schlüssel zu Kostenersparnis im Software-Engineering für die Automatisierungstechnik. TwinCAT 3 bietet auch hierfür die Plattform.

**Paulin:** Wir sehen eine zunehmende Sensibilisierung unserer Kunden auf eine gesamtheitliche Betrachtung der Kosten. Daher wird eine Betrachtung des gesamten Lebenszyklus von Maschinen und Anlagen immer wichtiger. B&R unterstützt diesen Trend, in dem wir sehr großen Wert auf die perfekte Abstimmung von Hard- und Software legen. Dabei sehen wir unter dem Begriff ‚Automatisierungssoftware‘ nicht nur die Entwicklungsumgebung, sondern auch die Software auf den Hardware-Modulen. Diese haben maßgeblichen Einfluss auf die Produktivität im laufenden Betrieb – dem größten Teil des Lebenszyklus. Ein optimales Zusammenspiel aus einer Hand maximiert die Produktivität der Maschine oder Anlage und sorgt so für nachhaltige Wettbewerbsvorteile für unsere Kunden.

**Wegner:** Heute muss der Anwender für die Lösung einer Automatisierungsaufgabe oft umfangreiches Wissen über das verwendete System haben. Zukünftig wird sich der Anwender mehr auf die eigentliche Problemstellung konzentrieren können. Durch die Automatisierung wiederkehrender Tätigkeiten, die Unterstützung von Software-Assistenten und die Verwendung von modularen Applikationskonzepten wird teure Engineeringzeit eingespart werden. Gute Beispiele für die einzelnen Disziplinen gibt es schon, allerdings wird zukünftig das Zusammenspiel der Werkzeuge disziplinübergreifend innerhalb der Suite erheblich verbessert werden. Elementar wird aber auch der Aspekt Offenheit einer systemspezifischen Suite sein. Hier ist eine optimale Anbindung von Spezial-Tools für die Elektroplanung und die Prozesssimulation immens wichtig, ohne die der Durchstich nicht gelingen wird.

**Weiler:** Zum einen steigt die Bedeutung von Analyse-Tools zur Messung und Steigerung von Effektivität und Effizienz automatisierter Anlagen bzw. der optimierten Nutzung bestehender Ressourcen zur Sicherung und Ausbau der Wettbewerbsfähigkeit. Wonderware-Software bietet gerade in diesem Bereich einen ganzheitlichen Ansatz. Auf Basis einer zentralen Plattform mit einheitlicher Entwicklungsumgebung werden Daten nicht nur erfasst, sondern in sinnvolle Informationen umgewandelt. Funktionale Module zur Visualisierung, Stillstandsanalyse, Rückverfolgung, Batchmanagement können ergänzt werden. Zum anderen wird das Thema ‚Vir-



Thorsten Weiler von Wonderware

tualisierung von Automatisierungssoftware‘ in naher Zukunft an Bedeutung gewinnen. Wonderware unterstützt hierbei bereits den Einsatz seiner Softwarelösungen in Verbindung mit VMWare ESX Server.

**Werner:** Automatisierungssoftware wird sich vor allem an der Effizienz messen müssen, mit der Applikationen erstellt, getestet und gewartet werden können. Eine Software-Suite wird die Anforderungen der Zukunft nicht erfüllen können. Bei immer komplexeren Maschinen und Anlagenstrukturen ist eine Integration verschiedenster Engineeringtools unumgänglich.

**Weyer:** Entwicklungsumgebungen – oder von mir aus auch Software-Suites – müssen zukünftig Plattform übergreifend einsetzbar sein. Wir werden schlussendlich eine übergreifend einsetzbare Entwicklungsumgebung für alle Hardware-Plattformen von Schneider Electric haben. Darüber hinaus müssen mit einer Entwicklungsumgebung alle Teilaspekte einer Automatisierungslösung bearbeitet werden können. Bei SoMachine Motion integrieren wir beispielsweise die Safety-Programmie-

rung und bieten Testwerkzeuge zur Verbesserung der Software-Qualität an. Die Konzentration aller Projektdaten in einem Tool übergreifenden, zentralen Daten-Pool ist ein wichtiger Punkt. Bei SoMachine Motion wird er beispielsweise die Basis für ein echtes Multi-User-Konzept sein. Und natürlich muss alles noch wesentlich komfortabler werden, der Weg wird für PacDrive 3 in Richtung einer echten grafischen Programmierung führen.

zg

#### INFO-TIPP

*Stimmen Sie als Automatisierungsanbieter oder -anwender mit den Expertenmeinungen überein? Wir freuen uns auf ein hoffentlich reges Feedback, per Mail an:*

• [ea.redaktion@konradin.de](mailto:ea.redaktion@konradin.de)  
oder nach einer einfachen Registrierung über das Online-Forum *Automatisierung – Fragen und Problemlösungen*:

• [www.wirautomatisierer.delforum](http://www.wirautomatisierer.delforum)