



IEC 61499 - Die neue IEC 61131?

**Valeriy Vyatkin, University of Auckland, New Zealand
Christopher Schemm, infoteam**

**Presented at Automation Day,
October, 17th, 2011
Nuremberg, Germany**

IEC 61499 - Die neue IEC 61131? (Is IEC 61499 a new IEC 61131-3?)

**Valeriy Vyatkin, University of Auckland, New Zealand
Christopher Schemm, infoteam**

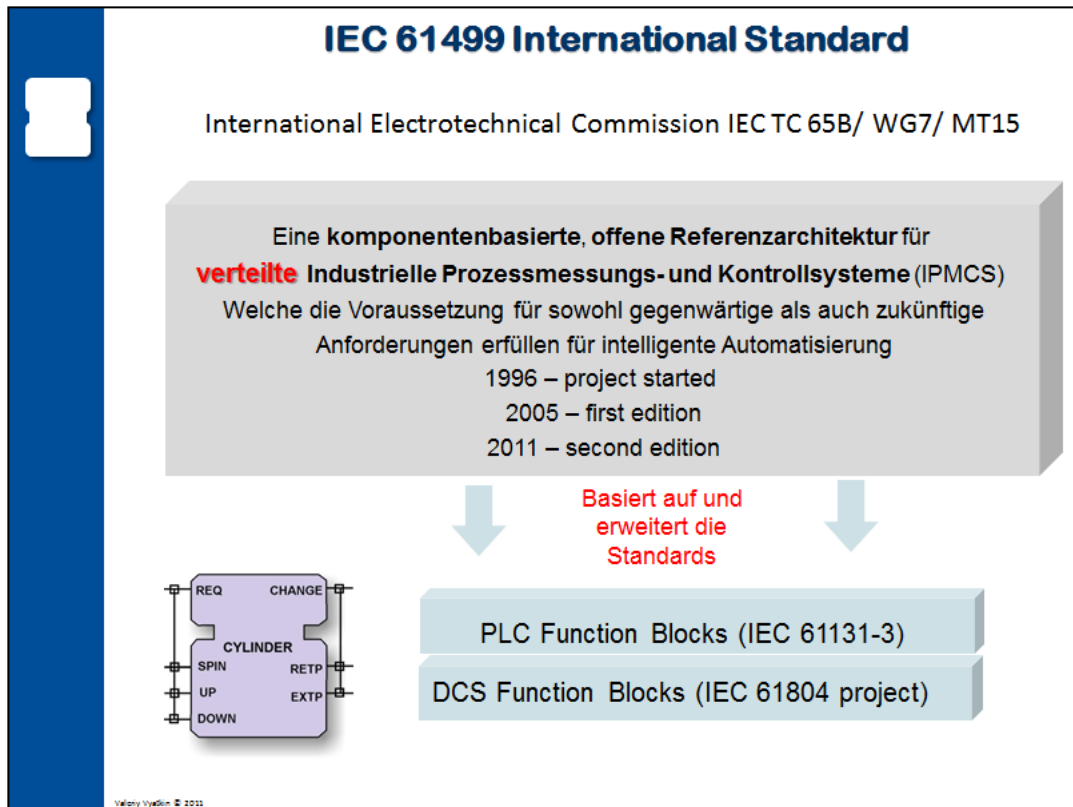
**Presented at Automation Day,
October, 17th, 2011
Nuremberg, Germany**

Plan

- Was ist die IEC 61499?
- Blockdiagramm-Denkweise
- Ein kurzes Einführungsbeispiel
- Ereignisgesteuertes Komponenten-Modell
- Schlüsselvorteile: Systemlevel-Design-Offenheit, Übertragbarkeit, Kompatibilität
- Tools und Plattformen
- Nutzen für den Endbenutzer: Illustrationen aus industriellen Fallstudien
 - Flexible Fertigung
 - Smart-Grid
- Unterschiede zur 61131-3
 - Wiederverwendung, Flexibilität, Verteilung, Systemlevel-Design
- Effizienz
 - Bedenken
 - Cyber-physical Design: Illustration der Flughafen-Gepäckabfertigung
- Aktivitäten und Schlussfolgerungen



© 2011 Valery Vyatkin - Original text and graphics; Christopher Schumm - Translation to German



IEC 61499 ist ein internationaler Standard, der eine Referenzarchitektur für verteilte Automatisierungssysteme beschreibt. Insofern unterscheidet er sich grundsätzlich vom bekannten IEC 61131-3 Standard, welcher eine Reihe von Programmiersprachen für die SPS-Programmierung beschreibt. Informell ist der Standard dafür bekannt, dass er ein komponentenbasiertes Modell verwendet, welches ähnlich zu Blockdiagrammen ist. Deshalb wird es oft als “Funktionsbausteine” bezeichnet.

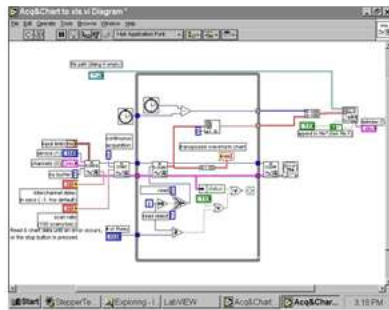
Das IEC-Projekt begann im Jahre 1996 und der Draft war 1999 fertiggestellt. Anschließend dauerte der industrielle Testprozess bis 2003. Schließlich wurde die erste Edition im Jahre 2005 veröffentlicht, die Zweite wird dieses Jahr erwartet.

Dieser Standard ist ziemlich umstritten: Trotz langer Vorgeschichte ist seine Akzeptanz in der Industrie sehr niedrig. Dennoch ist seine “akademische Akzeptanz” interessanterweise ziemlich hoch: es gibt hunderte von Veröffentlichungen, ein Dutzend Doktorarbeiten, einige Forschungsgruppen weltweit, von Portugal bis Neuseeland, die an den verwandten Fragen arbeiten.

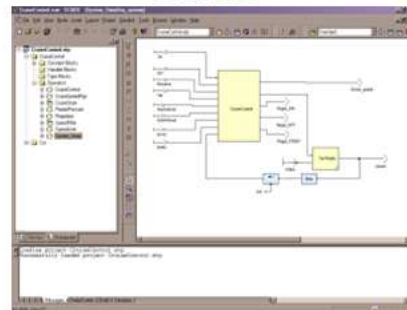
Das Schlüsselwort im Titel des Standards ist “VERTEILT”! Der Begriff “verteilte Automatisierung” wurde nachträglich auf drei Generationen von Technologie angewendet. Zuerst war es mit dem Aufkommen der Feldbereichsnetzwerke möglich, von Sensoren zu sammeln, welche über verstreute geografische Orte verteilt waren, während ihre Steuerung zentral mittels Speicherprogrammierbaren Steuerungen (SPS) durchgeführt wurde. Als Nächstes ergaben sich Versuche, SPS in Systeme zu integrieren, die über Netzwerk miteinander kommunizieren, indem man Integrationskomponenten-Architekturen einbrachte, wie Modbus-IDA und PROFInet-CBA. Zu guter Letzt, kommt es zur echten verteilten Automatisierungsentwicklung, wo die Intelligenz von Anfang an als dezentralisiert und eingebettet in Software-Komponenten entworfen wird, welche beliebig über vernetzte Hardware-Geräte verteilt werden können. Somit ist IEC 61499 eine “Sprache” für solche Systeme, bei welchen es wichtig ist, solche Stücke “verteilter Intelligenz” zusammenzufügen.

Blockdiagramm-basierte Programmierung

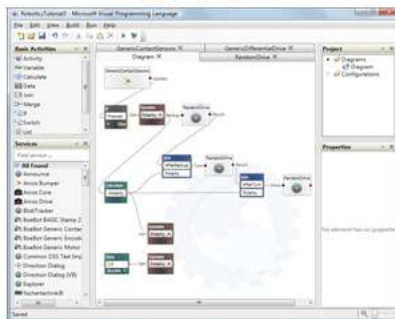
LabVIEW



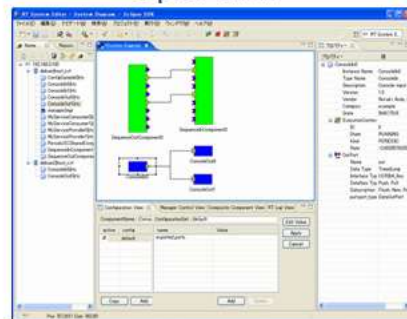
SCADE



MS VPL



Open RTM




© 2011, Valery Vladim - Original tool and graphics, Christopher Schomm - Transition to German

Das Blockdiagrammmodell zur Beschreibung von Kontrollsystemen ist nichts ungewöhnliches für Kontrollingenieure. Es gibt eine Vielzahl von kommerziellen Tools, die diese Denkweise unterstützen und nutzen. Um einige zu nennen: das FBD des IEC 61131-3 Standards, Simulink, SCADE und LabVIEW.

Auch das Modell des Zustandsautomaten ist weit verbreitet, besonders bei eingebetteten Systemen und in der Computerwissenschaft. Im Zusammenhang mit Automatisierung kennen wir alle SFC, sowie einige proprietäre Sprachen, z.B. von SIEMENS (HyGraf), oder Flussdiagrammsprachen von ISaGRAF und Nematron.

Hier sehen wir mehr Beispiele: LabVIEW, Microsoft Visual Programming Language, SCADE und Open RTM, benutzt für die Programmierung von Robotiksystemen.

Frühe Papiere mit IEC 61499-Ideen



 Control Eng. Practice, Vol. 4, No. 6, pp. 855-861, 1996
 Copyright © 1996 Elsevier Science Ltd
 Printed in Great Britain. All rights reserved.
 0967-0661/96 \$15.00 + 0.00

PII: S0967-0661(96)00078-0

**ASYNCHRONOUS AND SYNCHRONOUS APPROACHES FOR
 PROGRAMMING DISTRIBUTED CONTROL SYSTEMS BASED ON
 STANDARDS**

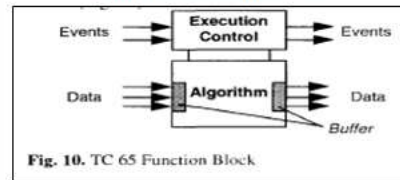
R. Schoop* and A. Streitzoff**

*AEG Schneider Automation, Steinheimer Strasse 117, 62580 Seligenstadt, Germany (schoop@redwin.de)
 **AEG Schneider Automation, One High Street, North Andover, MA 01845-2699, USA

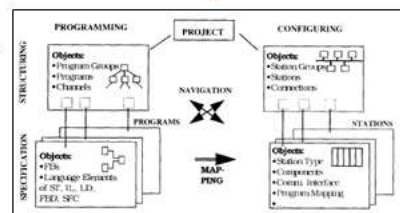
Abstract: Based on a general design model for distributed control systems, and using standardised languages of IEC 1131-3 for control, three approaches to programming are investigated. The first is based on IEC programs with extensions, the second is a decomposition of programs with SFC notations and the third approach uses function blocks corresponding to the IEC TC65 Function Block Standard. The approaches are specified and compared, and conclusions for their use and for further work are drawn. The intention of the contribution is to discuss possibilities for open programming models, rather than to present final results.

Keywords: Distributed control, distributed models, functional blocks, open control systems, programming approaches, sequential control, standards.

Komponentenmodell



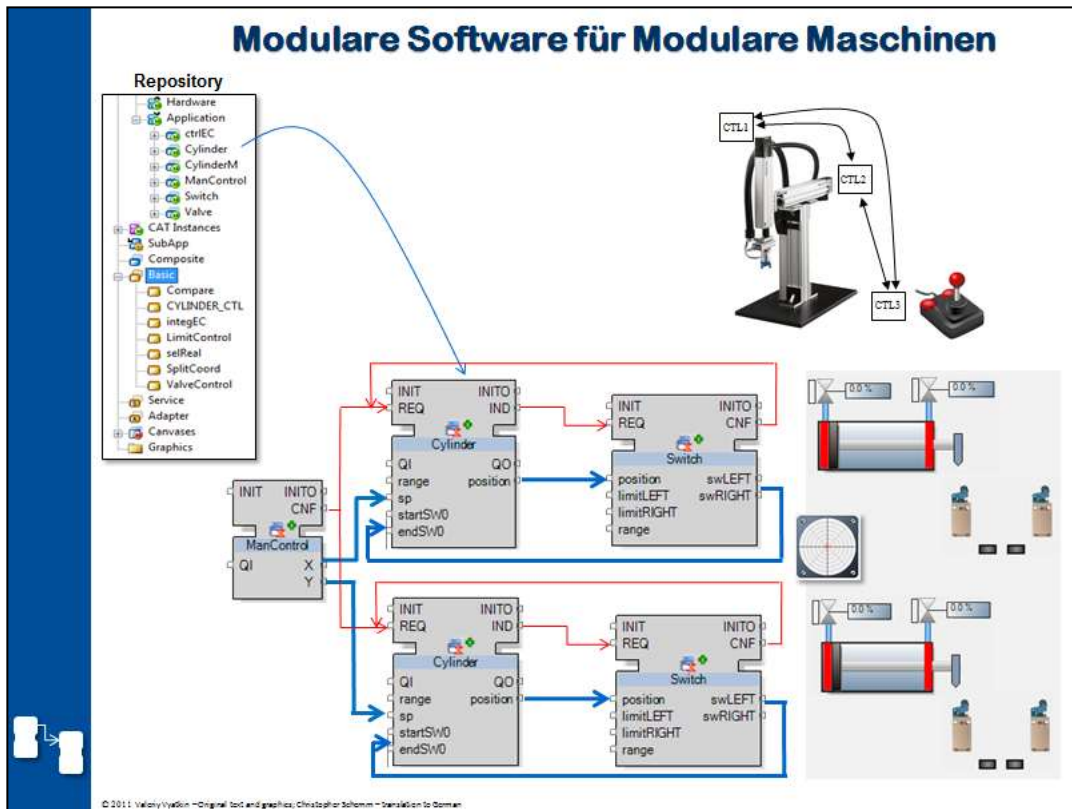
Verteilungsmodell



Wer hat den Standard entwickelt? Dieselbe Arbeitsgruppe der IEC, die auch IEC 61131-3 entwickelt hat. Hoch angesehene Experten in der Automatisierung. SIEMENS A&D: Hans-Peter Otto, Rockwell: Dr Jim Christensen; Schneider: Heinz-Dieter Ferling, GT: Antonio Valentini, Robert Lewis ...

Ich habe ein Forschungspapier entdeckt, geschrieben im Jahre 1995, mit Ideen, welche in IEC 61499 verwirklicht wurden. Einer der Autoren ist Dr. Ronald Schoop, derzeit CEO von Schneider Electric Automation Germany.

Insbesondere schlägt es vor: "Es wäre besonders gut, wenn dieses FB-Konzept ein Programmiermodell wäre..."



Hier ist ein (sehr einfaches) Beispiel für einen Entwicklungsprozess mit IEC 61499 zu führen. Angenommen wir wollen einen kleinen Setzpacker erstellen und diesen mit einem Joystick bedienen, um sicherzugehen, dass er sich schön ruhig bewegt.

Wenn man eine Maschine wie diese konstruiert, ist es eine gute Idee, einige Standard-Bausteine zu verwenden, die sogenannten "Mechatronischen Komponenten".

Kontrollarchitektur: Im einfachsten Fall kann dieses System bis zu drei vernetzte Kontrollpunkte beinhalten: Einen in jedem Zylinder und einen im Joystick. Aber natürlich kann es auch von einem einzelnen Kontrollgerät gesteuert werden. Die Wahl kann auf verschiedenen Prioritäten basieren. Was wir allerdings wollen, wenn die Steuerungssoftware entworfen ist, ist maximale Abstraktion von einem bestimmten Hardware-Layout.

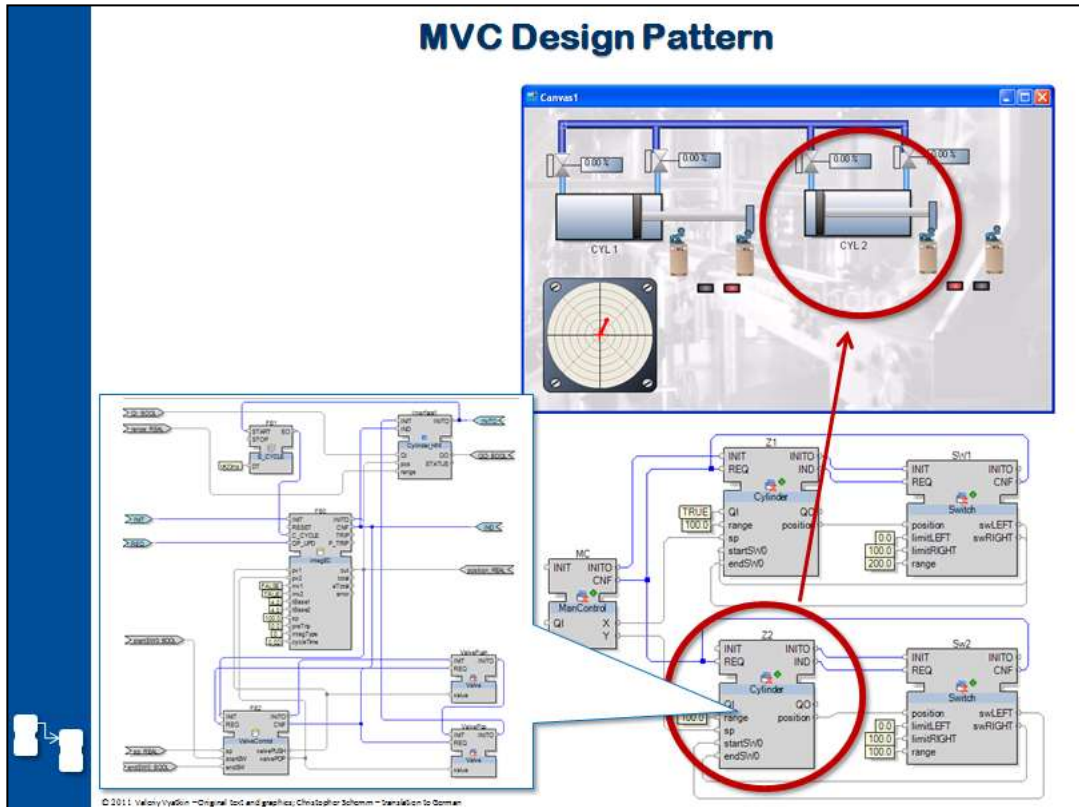
Wie Sie dem Diagramm entnehmen können, sind die beiden wichtigsten Komponenten dieses Manipulators "Zylinder". Eigentlich bestehen die Zylinder aus dem Schaft mit zwei integrierten Ventilen und zwei Positionssensoren.

Das Diagramm stellt die genaue Struktur beider mechatronischer Komponenten dar, auch wenn es kleine Unterschiede unter ihnen geben kann, erklärt durch verschiedene mechanische Eigenschaften, verschiedene Parameter der Zylinder oder Ventile.

Heutzutage ist es offensichtlich, dass Software in ähnlicher Weise aus Blöcken zusammengestellt werden kann, wie der mechanische Teil der Maschine. Der Software-Entwurf für dieses System beginnt damit, das Lager zu öffnen und die Software-Komponenten für Zylinder, den Sensorblock und den Joystick zu "instanzieren". Diese werden durch die Linien verbunden, welche Ereignisse und Datenfluss repräsentieren. Diese drei verbundenen Komponenten werden ausreichen, um eine Dimension des Manipulators zu kontrollieren.

Für die zweite Dimension müssen wir wortwörtlich ein paar weitere derselben Komponenten instanzieren.

MVC Design Pattern



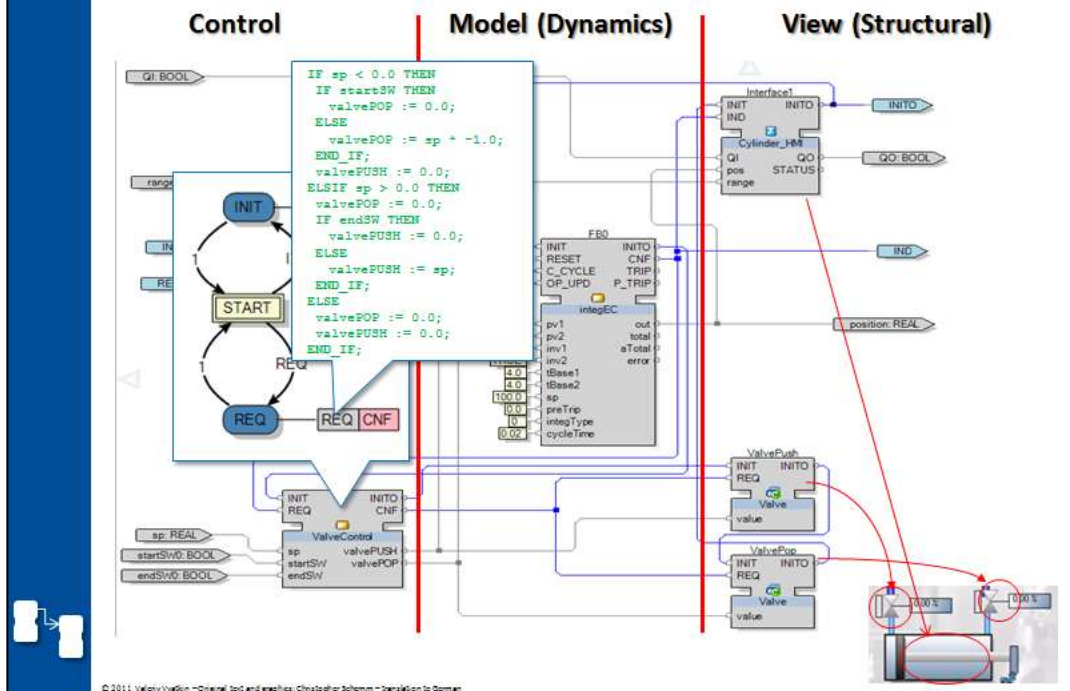
Das entworfene Netzwerk von Funktionsbausteinen ist eine vollständige, ausführbare Spezifikation dieses Systems.

Es kann nicht nur die Kontrollseite abdecken, sondern auch andere wichtige Funktionen wie Visualisierung, HMI und simulationsbasierte Tests.

Das Model-View-Control (MVC) Entwurfs-Pattern hat sich als äußerst nützlich erwiesen, um Software-Komponenten zu erschaffen, die alle diese Funktionen tragen können. Die Zylinderkomponente zum Beispiel kann sich um die Ausführung der Komponente im View panel kümmern, zusammen mit ihrer Steuerung, und kann sogar ihre Dynamik modellieren.

Dies wird erreicht durch die Spezifizierung dieser Komponente als Verbundfunktionsbaustein.

Zylinder: MVC Design



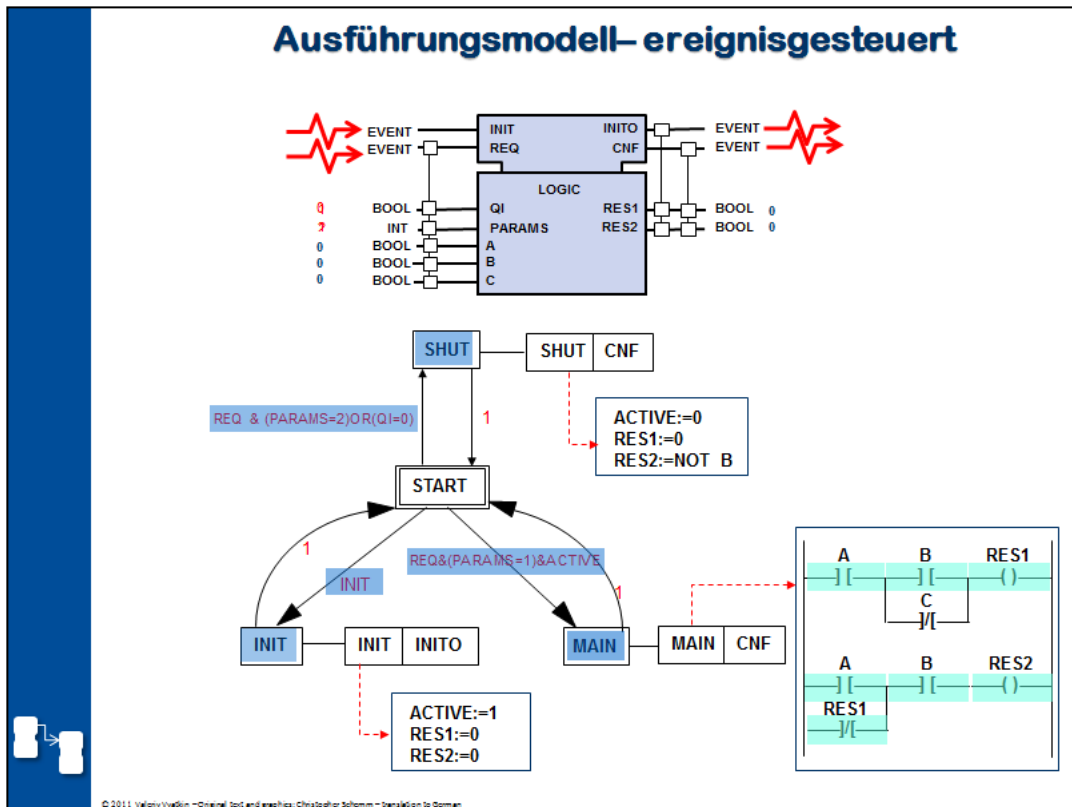
Schauen wir uns nun im Detail an, wie dies auf dem Level der Komponente erreicht wird. Man kann hier drei Gruppen von Komponenten unterscheiden. Der Kontrollteil wird repräsentiert durch den ValveControl Funktionsbaustein, welcher den Set Point Input Parameter “sp” annimmt und in zwei Kontrollsignale übersetzt, welche den Grad der Öffnung der zwei Ventile bestimmt, was den Zylinder in beide richtungen bewegt.

Diese Signale können direkt zu den Ventilen laufen, aber auch zum Modell der Zylinderdynamik, welche die Koordinaten des Schafts berechnet.

Dieser Parameter (vielleicht im realen System nicht messbar) kann verwendet werden, um die Position des Schafts und den status der Ventile zu visualisieren.

Was ist so toll daran, Modelle in Komponenten zu tragen? Man kann ganz eine Simulations- und Testumgebung für die Anwendung schaffen.

Was testen wir? Zuerst die Kontrolllogik. Schauen wir uns an, wie die Kontrolllogik in diesem Fall aussieht. Der Controller wird als Zustandsautomat dargestellt, welcher verschiedene Reaktionen auf das Initialisierungsereignis und das Parameter-Update beschreibt (diese werden von den Endpositionssensoren gelesen). Die Reaktion wird im Algorithmus beschrieben, welcher in jeder beliebigen SPS-Programmiersprache geschrieben werden kann (Gegenstand der Implementationsunterstützung). In unserem Beispiel ist der Code in ST geschrieben.

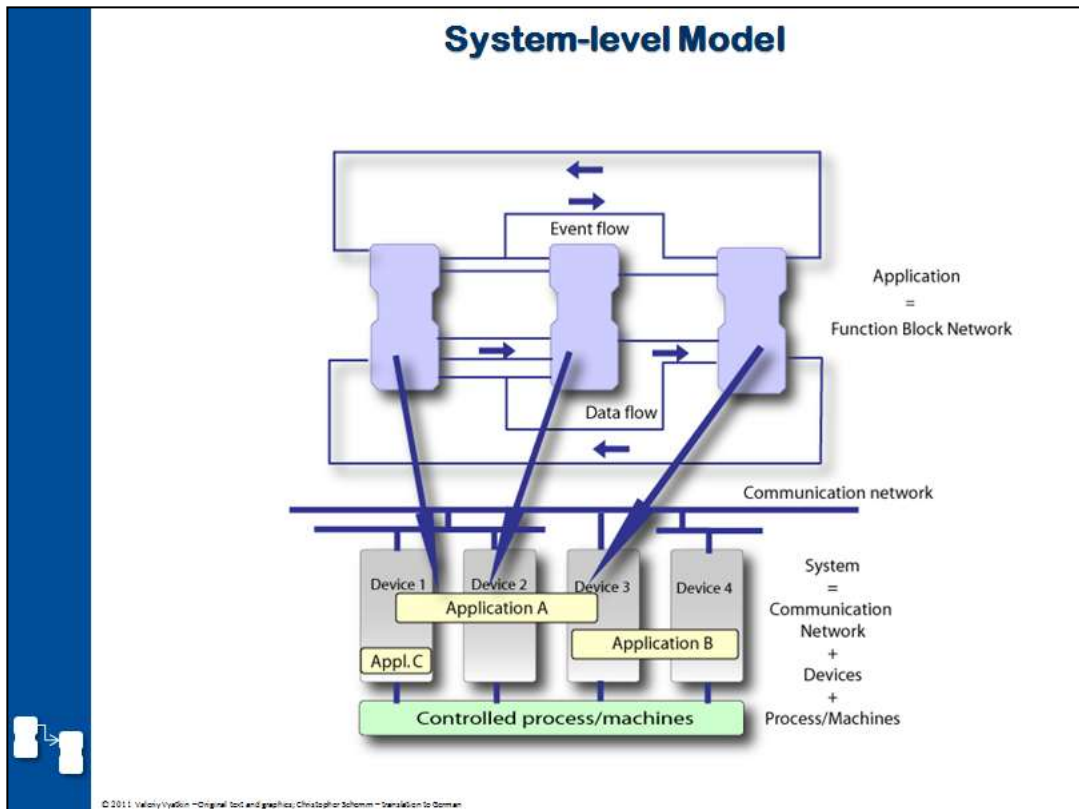


Betrachten wir ein Beispiel eines Funktionsbausteins, der drei verschiedene Reaktionen kombiniert:

1. Initialisierung: Setze die Outputs und die interne Variable ACTIVE auf ein paar gewünschte Werte, entsprechend dem Zustand “Working”;
2. Hauptfunktion: Behalte die Outputs berechnet als eine Funktion von Inputs und Outputs.
3. “Elegantes” Herunterfahren: Setze die Outputs und die interne Variable ACTIVE auf Werte entsprechend dem “Shut down”- Zustand.

Wir haben drei Algorithmen programmiert, die diesem Verhalten entsprechen, in Structured Text (für INIT und SHUT) und Ladder Logic für MAIN.

Am Anfang ist die Maschine im START-Zustand. Wenn das INIT-Ereignis ankommt, springt der Zustand der Maschine von START auf INIT.

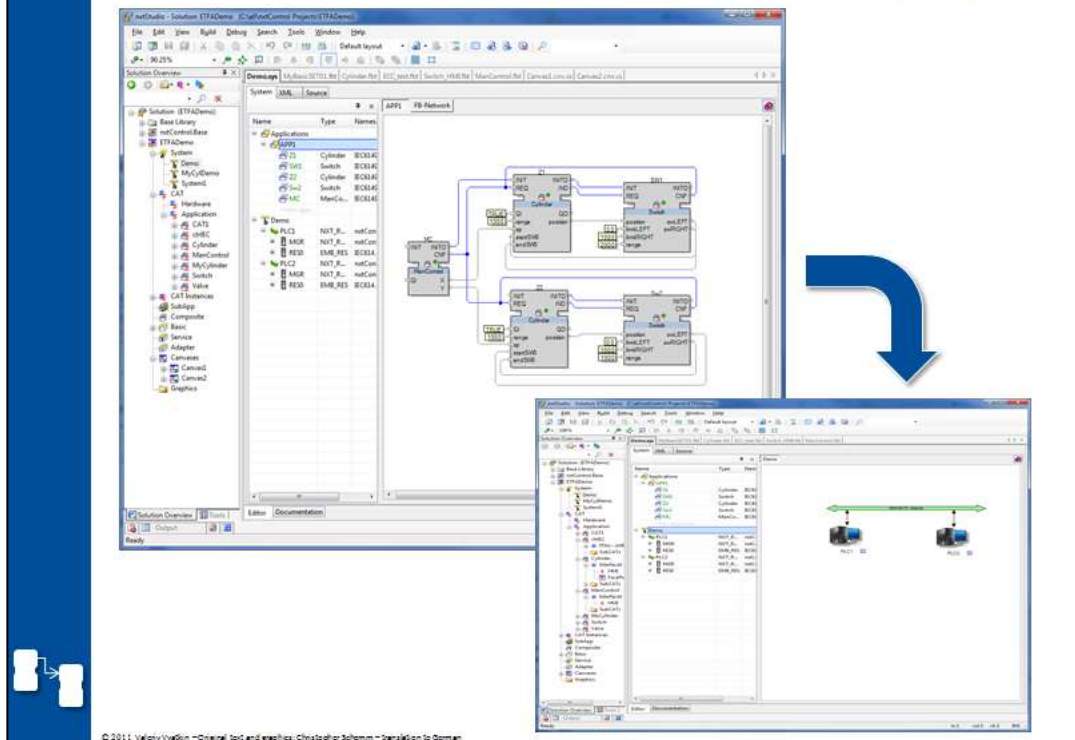


Ein großartiges Feature des IEC 61499 design flow ist die Trennung des Funktionalitätsentwurfs von der “Code zu Hardware”-Verbindung.

Der erste Schritt im Entwurf von verteilten Systemen ist die applikationsentwicklung (Funktionalität dessen, was erledigt werden muss).

Der zweite Schritt ist der Einsatz auf (vernetzten) Hardwaregeräten.

Systemmodell beinhaltet vernetzte Geräte

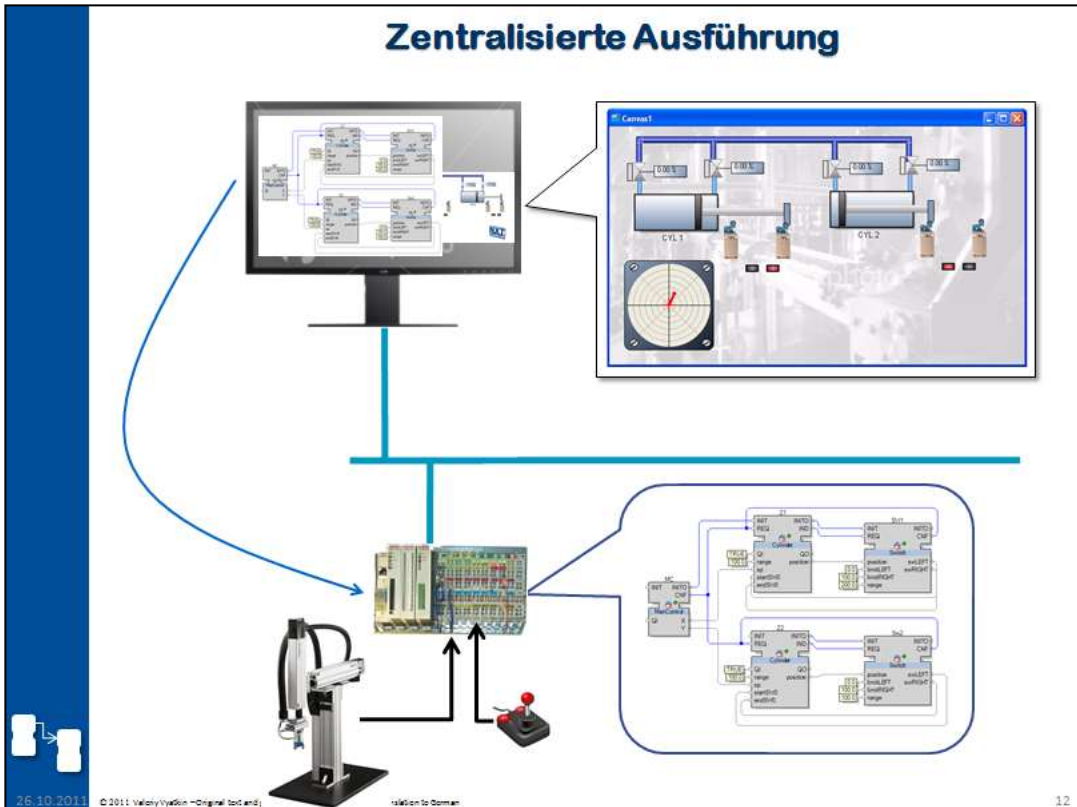


Was ist Verteilung?

Hier ist unsere Anwendung – Ein Netzwerk von Funktionsbausteinen, welches vollständig definiert, was zu tun ist und wie es zu tun ist.

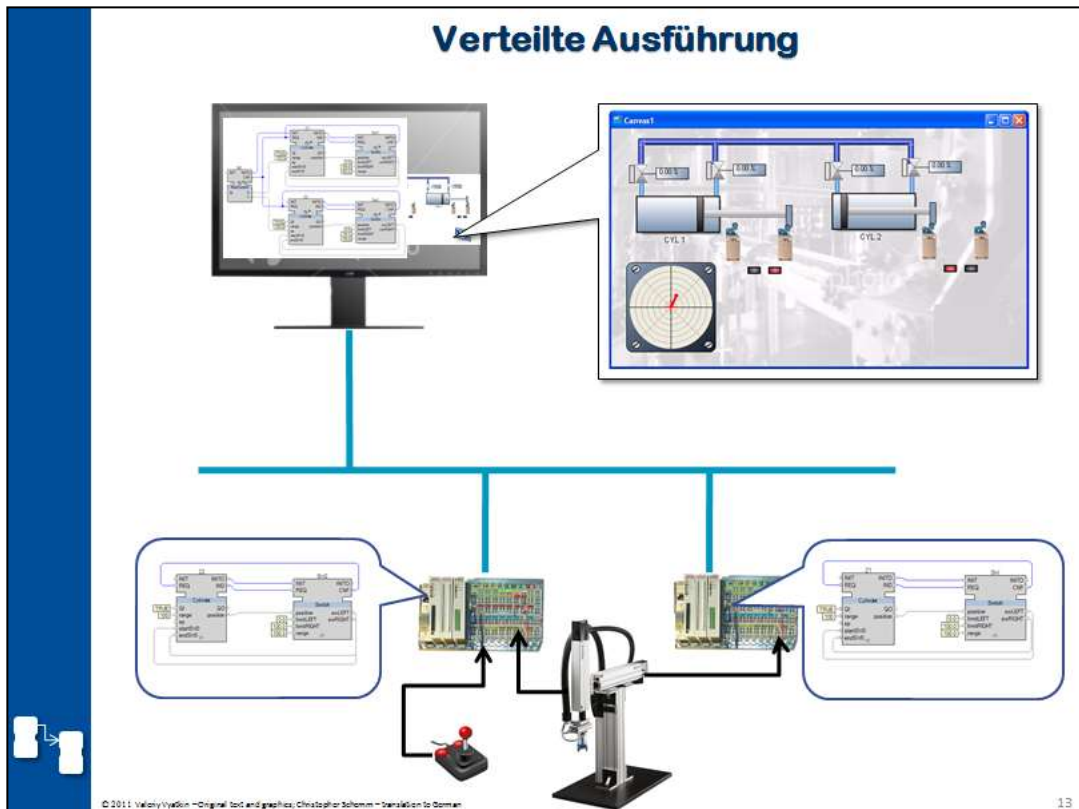
Aber es ist abstrakt, maschinenunabhängig. Nun müssen wir festlegen, welcher Funktionsbaustein in welchem Gerät arbeitet.

Diese *Verteilung* wird uns sagen, wie die Geräte miteinander kommunizieren müssen.



In der Automatisierung ist die Codeverteilung oft beeinflusst durch die Verbindungen zur Datenvermittlung. Wenn wir zum Beispiel alle Sensoren und Aktoren unseres Manipulators mit den I/O-Ports dieses einzelnen Controllers verbinden, macht es Sinn, die gesamte Anwendung auf diesem Kontrollgerät auszuführen. Visualisierung ist ein anderer Teil dieses verteilten Systems, sie wird auf dem PC laufen, der mit dem Controller über Ethernet verbunden ist. Diese Konfiguration ist zentralisiert, ungeachtet des komponentenbasierteC Software-Entwurfes.

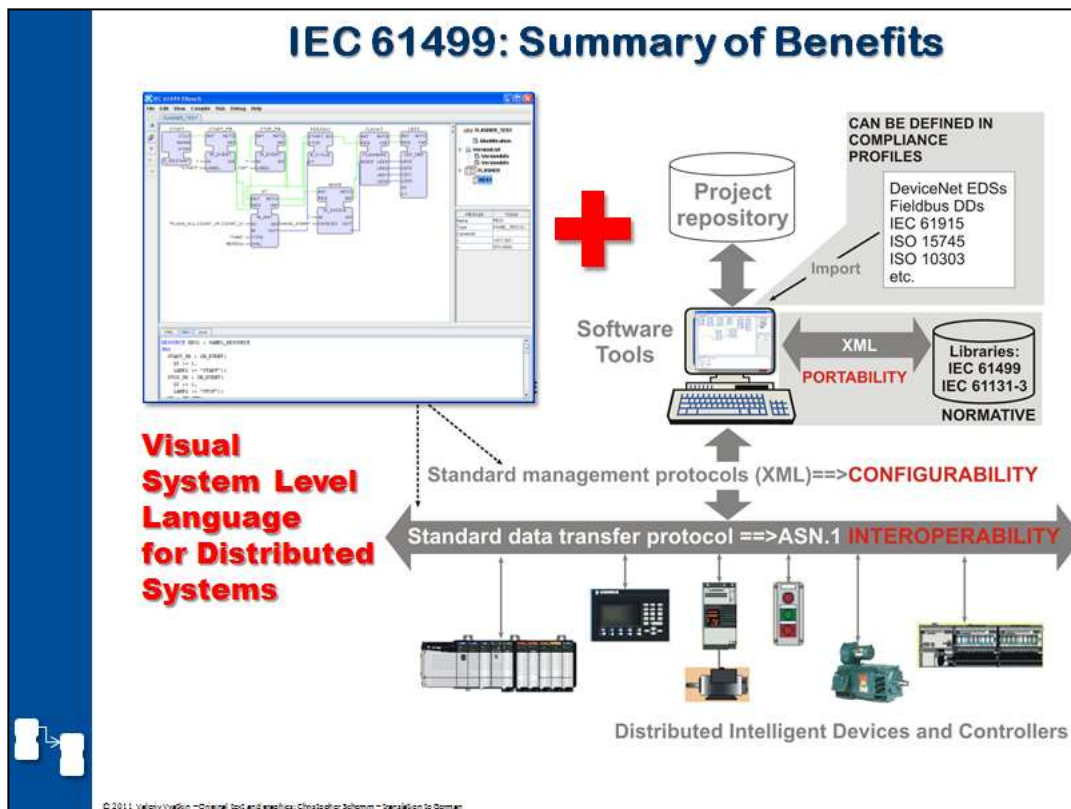
Die Datenfluss-Verbindungen zwischen den Komponenten könnte vielleicht in ein einfaches Datensharing über einen gewöhnlichen Speicher überführt werden.



Nun nehmen wir an, dass die Verbindung der Sensoren/Aktoren anders ist: Joystick und ein Zylinder sind mit einer Steuerung verbunden, aber der zweite Zylinder ist mit einer anderen Steuerung, ebenfalls Teil dieses Netzwerks, verbunden. Das Tool kann die Applikation automatisch auf diese zwei Geräte aufteilen. Wann immer eine Ereignis- oder Datenflusslinie die Gerätegrenzen überquert, wird das Tool Verbindungscode einfügen, um die Nachricht durch einen bestimmten Protokoll-Stack zu übermitteln.

Der Extremfall kann sein, alle drei Geräte als Prototypen zu verwenden.

Die Vorteile der Verwendung von Funktionsbausteinen sind hier: 1) die Möglichkeit, verteilte Logik zu simulieren; 2) Das automatische Einfügen des Kommunikationscodes durch das Tool.



Ein offenes *Funktionsbausteinmodell* stellt einen Mechanismus für die Kapselung von Automatisierungsfunktionen zur Verfügung, in einer mehr portierbaren Form, als es bisher möglich ist, zum Beispiel in SPS. Dies stellt die Kompatibilität von verteilten Automatisierungsapplikationen auf der Quellcodeebene sicher sowie ihre äquivalente Ausführungssemantik, auch auf unterschiedlichen Plattformen.

Der offene Gerätemanagement-Mechanismus von IEC 61499 stellt *Konfigurierbarkeit* von konformen Geräten durch konforme Software-Tools zur Verfügung.

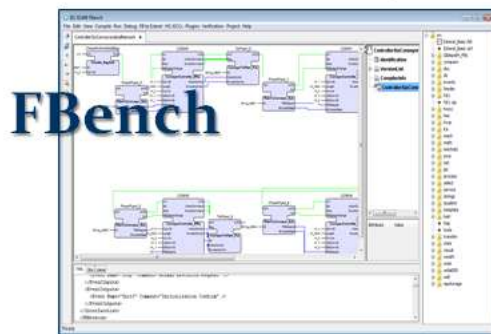
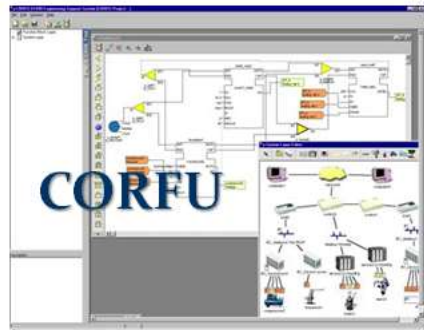
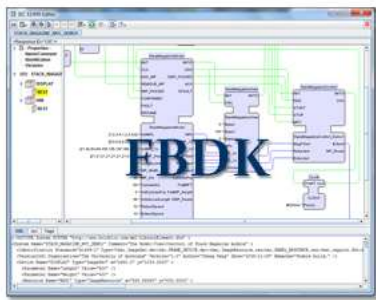
Funktionsbausteine von Dienstschnittstellen stellen einen Mechanismus für die Kapselung von hardwareabhängigen Funktionen in dieselben Sprachkonstrukte wie die der restlichen Applikation zur Verfügung.

Teil 2 des Standards definiert das XML-basierte Datenformat von Funktionsbausteinen, welches die Kompatibilität auch auf der Ebene der Quelldateien sicherstellt.

Bestimmte Protokolle der Gerätekonfiguration oder des –managements können als Konformitätsprofile für den Standard definiert werden. Teil 4 des Standards definiert die Regeln, wie solche Konformitätsprofile organisiert werden müssen.

Das ASN.1-Protokoll ermöglicht plattformunabhängige Datendarstellung.

Experimental Tools



Den größten Einfluss hatte FBDK, entwickelt von Rockwell's Dr. James Christensen. FBDK basiert auf Java.

Dies war die erste "Referenz"-Implementierung.

Das CORFU-Tool, entwickelt von Prof. K. Thramboulidis in Patras, Griechenland, stellt eine Brücke zur UML dar.

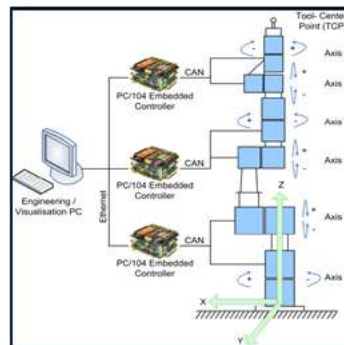
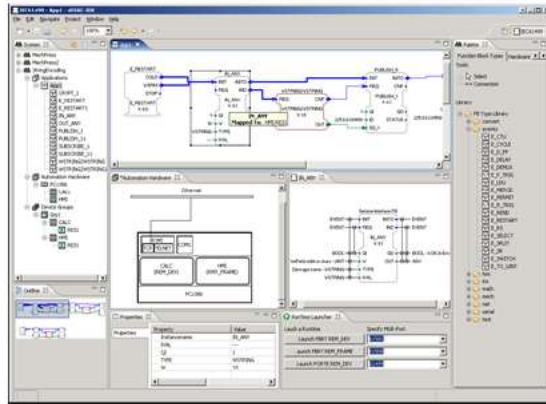
Fbench war das erste Open-Source-Projekt, geführt vom Autor dieses Vortrages. Es basierte ebenfalls auf Java.

Eine Zahl von Hardwaregeräten wurde für Experimente verwendet, z.B. Java-basierte controller von Elsisit, TCS-NZ, etc.

Diese Tools waren extrem nützlich für die Forschungsgemeinschaft.

4DIAC IDE und RTE (forte)

4DIAC-IDE



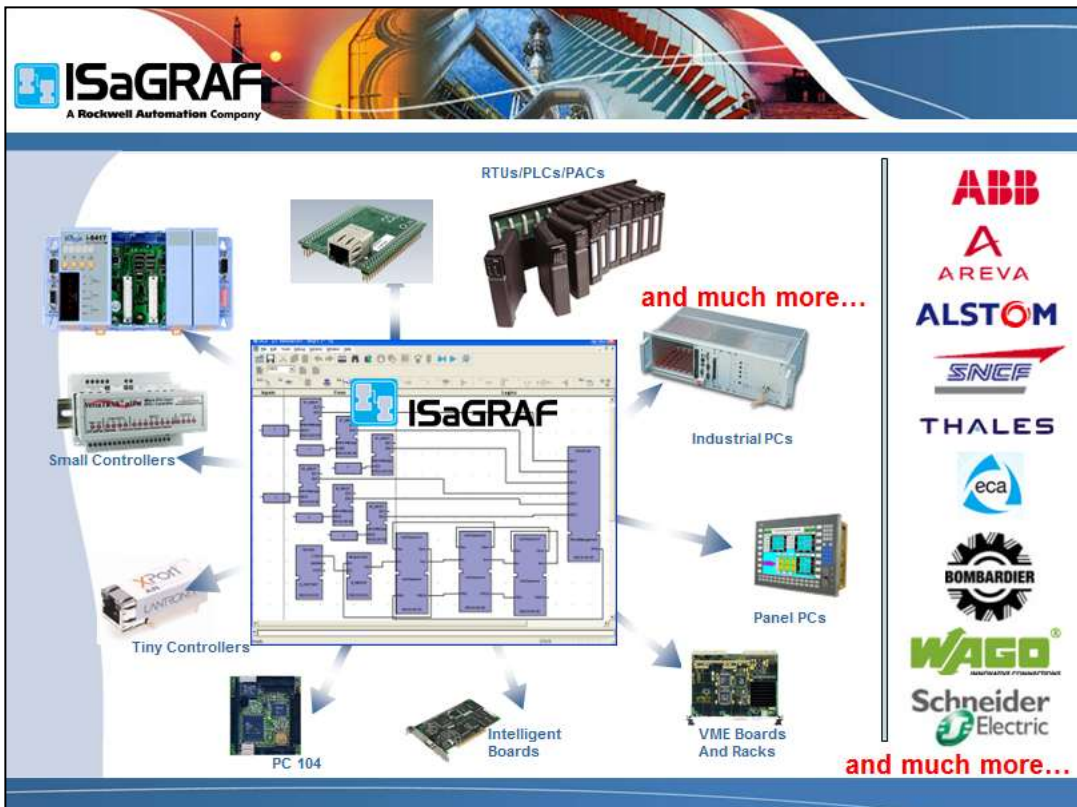
© 2011 Valeriy Vykhod - Original: toc and graphics; Christopher Schömm - translation to German

Dieses Open-Source-Projekt, geführt von der TU Wien, hat starken Einfluss auf kommerzielle entwicklungen wie NxtControl.

Das Projekt zielt auf die Entwicklung einer IDE ab, programmiert in Java unter Eclipse, und einer Laufzeitumgebung programmiert in C, und somit portierbar auf die meisten eingebetten Plattformen.

This is a lot more systematic development with \$2.7 M budget spent.

Es gibt viele Hardware-Plattformen, auf die 4DIAC-RTE portiert wurde. Die hier gezeigte zeigt einen 6 DOF-Roboter, gebaut von PROFACTOR. Das Schöne an diesem Roboter ist, dass er von einem heterogenen Netzwerk von drei CAN-verbundenen Kontrollgeräten gesteuert wird (PC/104 Embedded Controller).



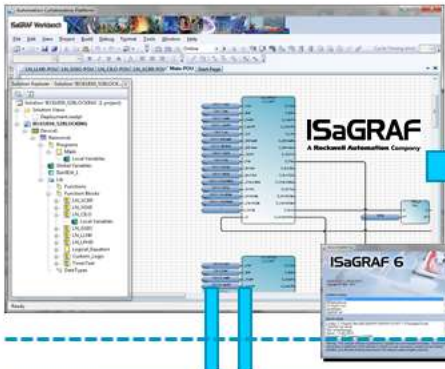
ISaGRAF ist ein Beispiel eines ausgereiften Automatisierungstools.

ISaGRAF (seit 2008 ein Rockwell Automation-Unternehmen) kombiniert die Unterstützung beider Standards IEC 61131-3 und IEC 61499, es ist also möglich, verteilte Kontrollanwendungen zusammen mit der Benutzung von SPS-Sprachen zu entwickeln. Dieses Tool unterstützt einen sehr nahtlosen Weg zur Code-Verteilung auf die Netzwerkgeräte. Die Werkbank fügt automatisch Kommunikationscode ein, wo er benötigt wird, während der Benutzer nur das globale Bild der gesamten verteilten Applikation sieht.

Die ISaGRAF Laufzeitplattform wurde auf Hunderte von Hardware-Plattformen portiert. Die Kundenliste schließt auch Hunderte von bekannten Firmen ein. Einige der "großen Namen" sind auf der rechten Seite der Folie aufgelistet.

Allerdings muss man zugeben, dass nur wenige Kunden von ISaGRAF die Vorzüge von IEC 61499 erkannt haben.

ISaGRAF v.6.0




ISaGRAF
A Rockwell Automation Company

ISaGRAF 6

ISaGRAF v.5 supports IEC 61499 since 2005. The next generation ISaGRAF v.6 released in 2010.

And, since 2008 ISaGRAF is Rockwell Automation company.

Allen-Bradley Micro800™ Family of PLCs
Featuring Allen-Bradley Connected Components Workbench™ Programming and Configuration Software



Connected Components Workbench

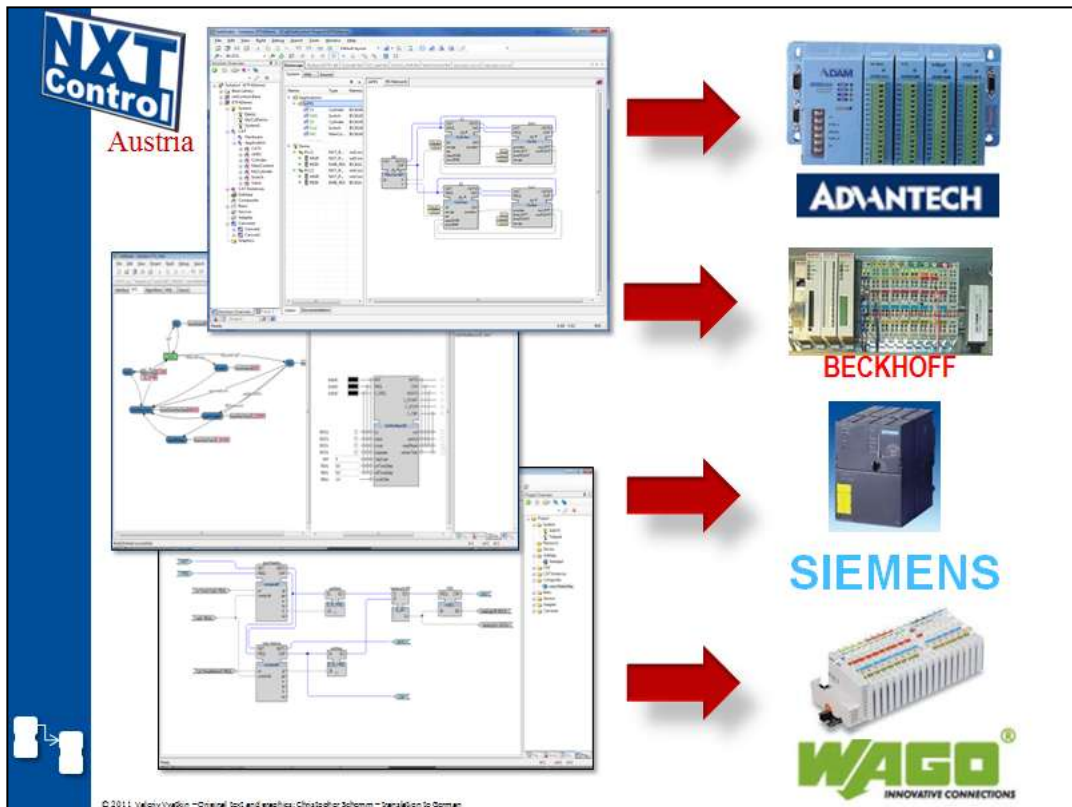
© 2011 Valeriy Vlodin - Original tool and graphics; Christopher Solomon - Transition to Don

Hier ist ein interessantes Beispiel der Benutzung von ISaGRAF.

ISaGRAF v.5 unterstützt IEC 61499 seit 2005. Die nächste Generation ISaGRAF v.6 wurde 2010 freigegeben.

Seitdem ISaGRAF zur Rockwell Automation Company gehört, kann man fragen, was der Einfluss dieser Akquisition auf sein Hauptgeschäft ist.

Wir haben keine genauen Informationen über die Rockwell-Pläne, aber in 2011 hat es eine neue Serie von SPS veröffentlicht, Micro800 genannt. Sie ist mit einem neuen Tool programmiert, Connected Components workbench genannt. Sogar ein ungeübtes Auge bemerkt erstaunliche Ähnlichkeit dieses Tools mit der ISaGRAF workbench v.6.



NxtStudio, entwickelt von der österreichischen Firma nxCtControl, integriert einen Ansatz zur verteilten Kontrolle basierend auf IEC61499 mit SCADA. Es ist eine Industrieklasse-Entwicklungsumgebung, welche das Design von Kontrollapplikationen und Visualisierung zusammen in einem Tool unterstützt. Dieser Ansatz hat große Vorteile in der Produktivität und Wiederverwendung sowohl von Kontroll- als auch von Visualisierungskomponenten. Einige Features von NxtStudio wurden lange von der IEC 61499 erwartet, zum Beispiel die Debugging- und Online-Ansichts-Infrastruktur, welche es erlaubt, aus der Ferne sowohl einzelne FBs als auch voll verteilte Applikationen zu debuggen. Ein weiteres Feature ist die automatische Generierung der Kommunikation während des Verteilungsprozesses der Applikation. Dies reduziert den Aufwand beim Entwurf von verteilten Kontrollanwendungen erheblich.

NxtStudio verlässt sich auf eine angepasste und optimierte Version der Forte Laufzeitumgebung. Es wurde auf eine Reihe von Hardwareplattformen portiert, wie programmierbare Automatisierungscontroller (PACs) von Wago, Beckhoff, Siemens MEC 31, Advantech und FESTO.

Mit diesen kraftvollen Tools, die sich bereits auf dem Markt befinden, haben die Entwickler eine genügende Auswahl, um die Vorzüge von IEC 61499 zu testen.

Anwendung : Molekulare Transportlinie

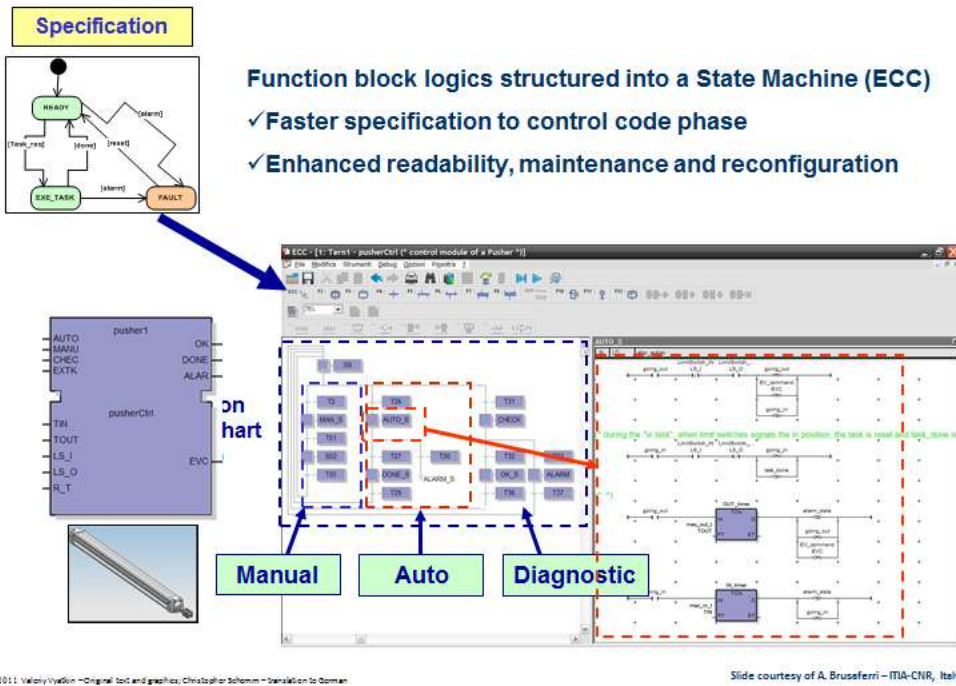
- Composed of 6 Terns made of rotating tables
- Each table integrate one rotation device and one to many pushing devices.
- Different optimized workpieces flows

© 2011, Valeriy Vlodkin - Original: www.valeriyvlodkin.com / Igor Schemm - translation to German Slide courtesy of A. Brussetti - ITIA-CNR, Italy

Das erste interessante Beispiel, das in diesem Video gezeigt wird, ist eine experimentelle Schuhfabrik nahe Mailand in Italien. Die Fabrik wurde entwickelt, um das revolutionär hohe Maß an Herstellungsflexibilität zu erreichen, dank welcher individuell maßgeschneiderte Schuhe mit den gleichen Kosten hergestellt werden könnten wie für massenproduzierte.

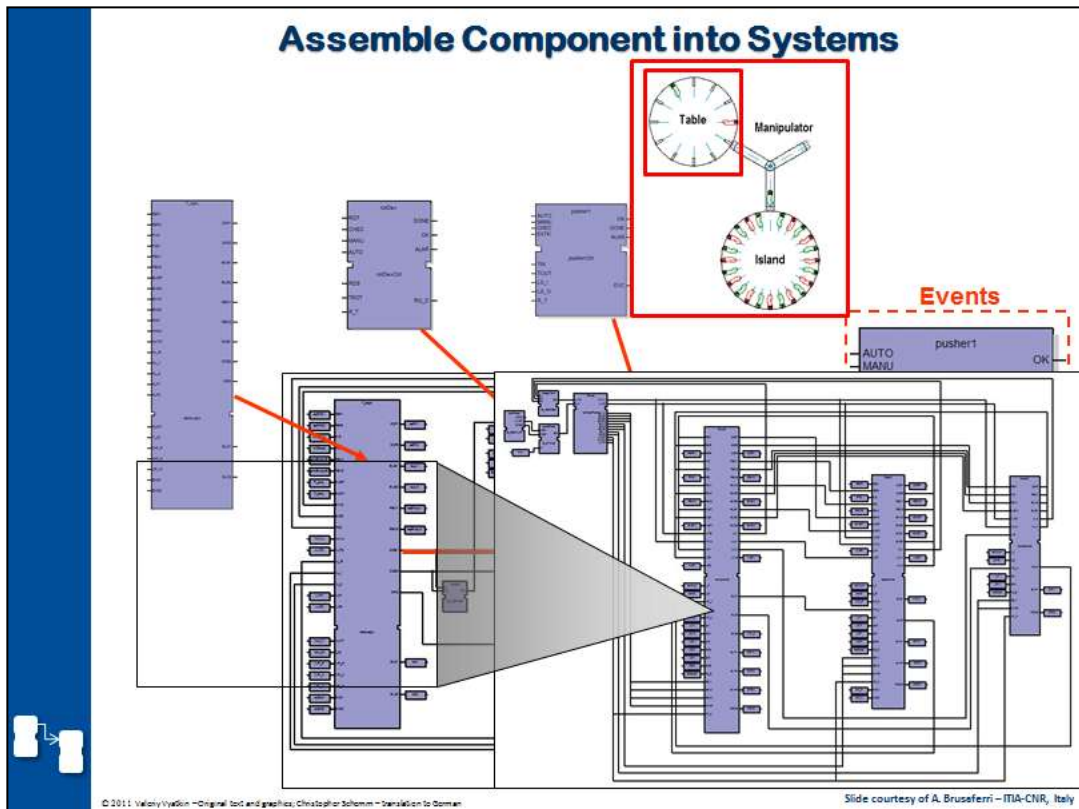
Für diesen Zweck wurden neue Arten von materialbehandelnden Geräten, genannt “terns”, entwickelt. Die “terns” bilden einfach zu konfigurierende Linien, wegen ihrer Form “Molekular” genannt. Wie in der Abbildung zu sehen, erlaubt diese neue Ausrüstung, mehrere Produktwege durch das System zu beherbergen, in welchem jeder Weg zu einer bestimmten Abfolge von Operationen gehört.

Control Software Development



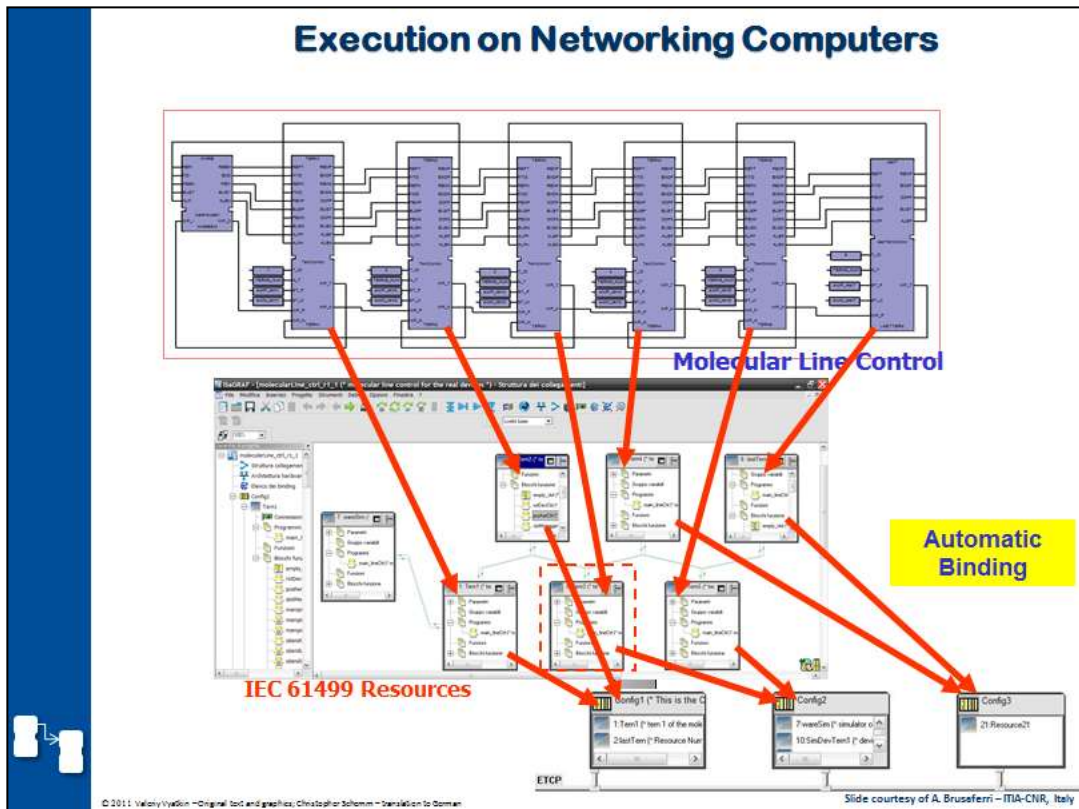
IEC 61499 erlaubt es, diese Zustandsautomaten in einen Funktionsbaustein zu kapseln, und zwar auf eine höchst bequeme Art und Weise, welche manuelle, automatische und diagnostische Modi, sowie Algorithmen, geschrieben in FBD oder Ladder-Logik kombiniert, was sich positiv auf die Wartung des Systems auswirkt.

Hier sehen wir das Beispiel der eingebetteten Code-Implementierung des Schiebers.



Anschließend werden Manager, Rotations- und Schieber-Steuerung zusammen in einem Drehtisch-Funktionsbaustein aneinandergesetzt. Man kann sehen, dass es eine klare Ereigniskette zwischen dem Manager, der einen Befehl gibt, und den "Slaves" (Rotation und Schieber) gibt. Diese Kette hilft, die komplexe Logik zu debuggen.

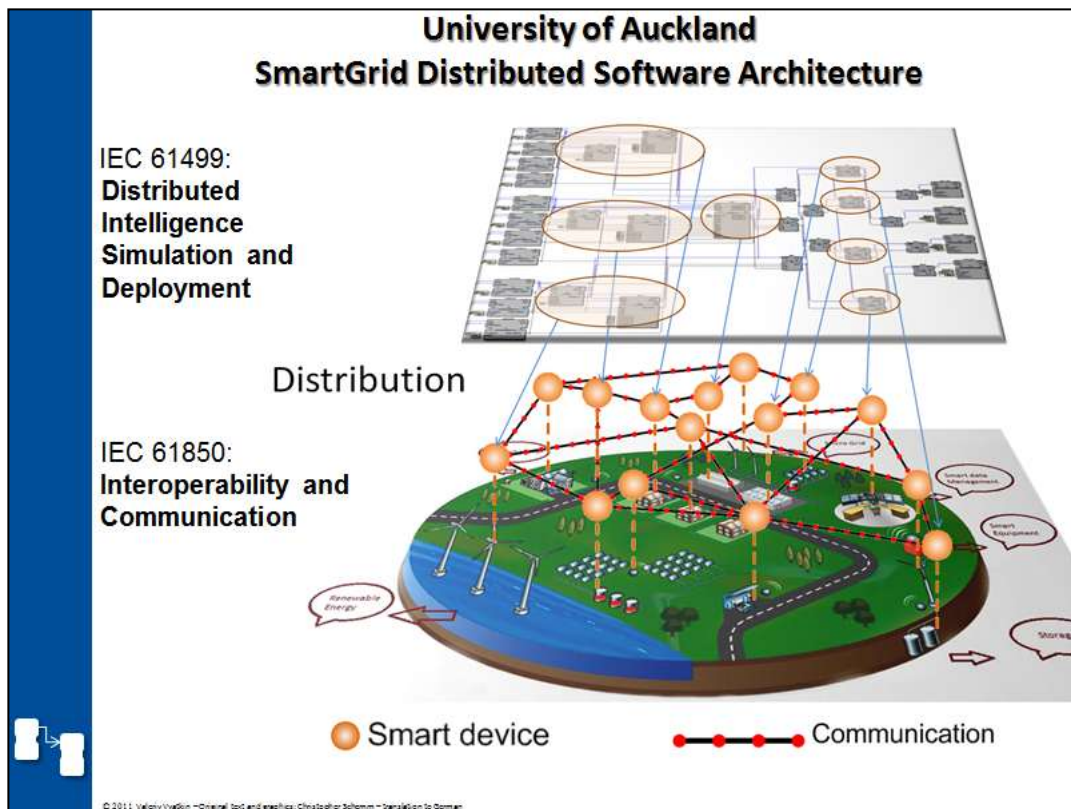
Wie man am Beispiel der Schieber-Steuerung sehen kann, hat der Baustein eine Ereignis-Schnittstelle, in welcher Eingangs-Ereignisse, ausgesendet als Befehle vom "Meister", zu verschiedenen Operationsmodi gehören.



Die Steuerung der Molekularlinie besteht aus sechs Instanzen des *TernControl*– Funktionsbausteins, von denen jeder aus drei Steuerungsfunktionsbausteinen besteht, welche verschiedene Teile der “tern” darstellen: *Table*, *Manipulator* und *Island*.

Dieser Grad des Bausteinkonzepts vereinfacht das Rekonfigurationspotenzial des Herstellungsprozesses erheblich. Jede physikalische Rekonfiguration der Anlagenmaschinerie führt zu einer Rekonfiguration von Software-Modulen, die alle nach der selben Vorlage gebaut und leicht ersetzbar sind.

Erst nachdem der gesamte code assembliert ist, muss der Entwickler eine Entscheidung bezüglich der Hardware-Verbindung treffen. Der Entwickler kann zuerst Funktionsbausteine in virtuelle Container “mappen”, Ressourcen genannt, und anschließend die Ressourcen auf die Geräte mappen. Dieses “Mapping” kann leicht verändert werden und wird von den Tools voll unterstützt.

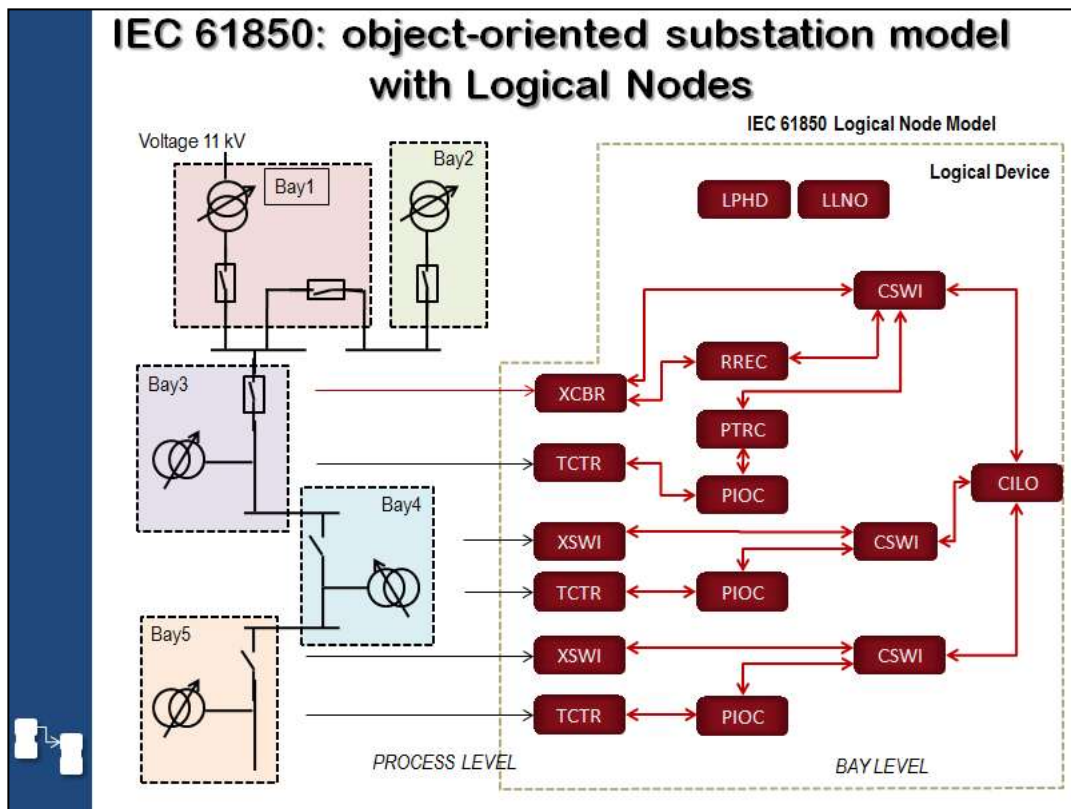


Ein weiteres Anwendungsgebiet, welches nach verteilter Steuerung verlangt, ist SmartGrid. Aus meiner Sicht demonstriert dieses Anwendungsfeld große Vorzüge der IEC 61499-Architektur in Sachen Entwurfseffizienz, aber auch ihre Fähigkeit, cyber-physikalische System-Herausforderungen, wie erklärt durch den Kollegin Professor Vogel-Heuser.

Wir an der Auckland-Universität entwickeln eine neue Automatisierungsarchitektur für SmartGrid basierend auf IEC 61499, kombiniert mit dem neuen IEC 61850-Standard für Kommunikationsnetzwerke und Systeme in Nebenstationen.

SmartGrid ist definiert als eine Integration von Elektrizität und Kommunikation, sodass elektrische Netzwerke “immer verfügbar, aktiv, interaktiv, verbunden und eng gebunden an die Kommunikation in einem komplexen Energie- und Informations-Echtzeitnetzwerk” sind. Das Ergebnis sind effizientere Energiesysteme, welche besser den wachsenden Energieverbrauch verwalten, Fehlerelastizität bereitstellen und nahtlos verteilte, erneuerbare Energien wie Wind- und Solarenergie integrieren.

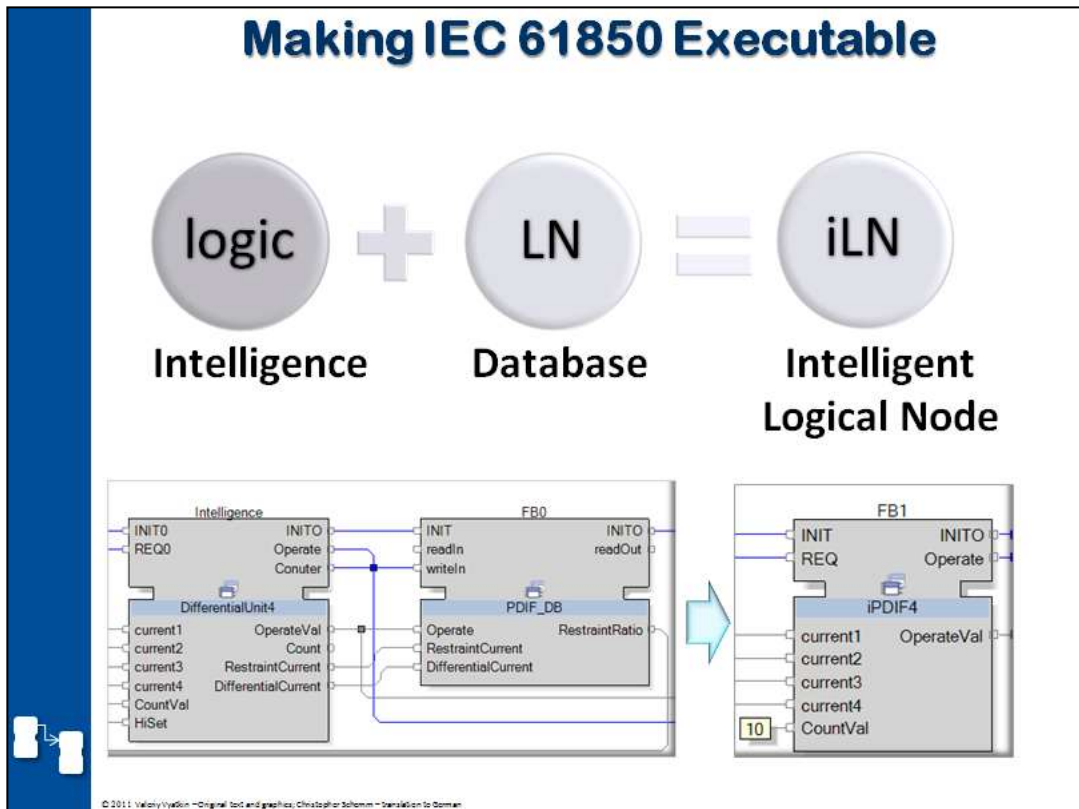
Es ist generelles Verständnis in der Forschungsgemeinschaft, dass traditionelle hierarchische Ansätze zum Automatisierungsentwurf dort nicht funktionieren. Die Kontrollarchitektur von SmartGrid wird von vielen Forschern als ein heterogenes Netzwerk von Controllern angesehen, welche in Punkt-zu-Punkt-Manier kommunizieren. Dieser Ansatz wird unterstützt durch den populären Standard IEC 61850.



IEC 61850 zerlegt Energie-Schaltanlagen, einschließlich Funktionen zur Überwachung, Kontrolle und zum Schutz, und primäre Geräte in Objekte und erhält somit eine objektorientierte Darstellung des Energiesystems. Daten und Datenattribute sind die Informationsmodelle für die Automatisierungsfunktionen und primären Geräte, welche in ein Set eingepackt und als Logische Nodes (LN) dargestellt werden. Somit kann die gesamte Anlage als ein Aggregat von logischen Geräten, bestückt mit logischen Nodes, modelliert werden, um bestimmte Funktionen auszuführen. Eine solche Darstellung strukturiert das Energiesystem und standardisiert die Informationssammlung und den -austausch, was Zusammenarbeit und Kompatibilität zwischen Produkten unterschiedlicher Hersteller bietet, die in einem Energiesystem verwendet werden.

Wie in der Grafik gezeigt, kann das Diagramm der Schaltanlage als Quelle ihres verteilten Softwaremodells verwendet werden, dargestellt als ein Set von logischen Nodes.

Das kleinste Objekt ist das Dateiattribut, welches in ein "Common Data"-Objekt gekapselt ist. Dies sind die Daten, die von Geräten und verwendet werden, wenn sie operieren.



Ein Defizit des IEC 61850-Standards ist, dass seine Strukturen statisch sind, er bietet keine Mechanismen, um verteilte Logik ähnlich der logischen Nodes zu definieren.

Wir haben versucht, dies zu beheben, indem wir eine neue “intelligente logische Node”-Architektur aufstellen. In dieser Architektur wird für jede LN der IEC 61850 ein intelligenter LN-Typ als Funktionsbaustein implementiert, mit Diensten, welche als Ereigniseingaben des Funktionsbausteins dargestellt werden.

Jede iLN ist modelliert als ein zusammengefüger Funktionsbaustein, wie in der Abbildung am Beispiel der differentialen Schutzeinheit (iPDIF) gezeigt: intelligente LN für differentialen Schutz. Wie jeder iLN besteht iPDIF aus einer Datenbasis und Intelligenz – Logik. DB ist die Sammlung von notwendigen und optionalen Daten nach IEC 61850, und führt Standarddienste anhand dieser Daten aus. Logik oder Intelligenz ist das benutzerdefinierte Verhalten dieses Funktionsbausteins.

FREEDM Green Hub Architecture

SmartGrid Intelligence Engineering:

- How to design and validate distributed intelligence?
- Where and how to deploy it?

FREEDM Industrial partners

NC STATE UNIVERSITY

ASU ARIZONA STATE UNIVERSITY

FLORIDA A&M UNIVERSITY

MISSOURI S&T

FLORIDA STATE UNIVERSITY

RWTH AACHEN UNIVERSITY

ETH

THE UNIVERSITY OF AUCKLAND

© 2011, Valley Vision - Original text and graphics: Christopher Sellen - Transition to German

Ich möchte Erfahrungen teilen, welche mit dem FREEDM-Programm in den USA gesammelt wurden, gegründet von der National Science Foundation. Das Programm wird gelenkt vom FREEDM Centre at North Carolina State University und integriert 30 Forschungsprojekte, welche in einer Reihe von amerikanischen, europäischen und Neuseeländischen Universitäten durchgeführt werden. Eigentlich sind dort drei ausländische partner: ZTH, RWTH und die Universität von Auckland, und nur diese wurde direkt aus dem Budget des Programms finanziert. Die Liste der Industriepartner von FREEDM ist beeindruckend. Das Projekt ist für 10 Jahre geplant und ist zurzeit im 3. Jahr.

Das Ziel des Programms ist es, ein funktionierendes Microgrid der nächsten Systemgeneration zu entwickeln, welches die einfache Integration von erneuerbaren Energien, Selbstheilung und Anpassung an Umweltveränderungen zeigt. Die Kerntechnologie von FREEDM ist das Konzept des "Green Energy Hubs": Ein Plug-And-Play-Gerät für Energielenkung, basierend auf einem neuen Typen eines cleveren "Solid state Transformers".

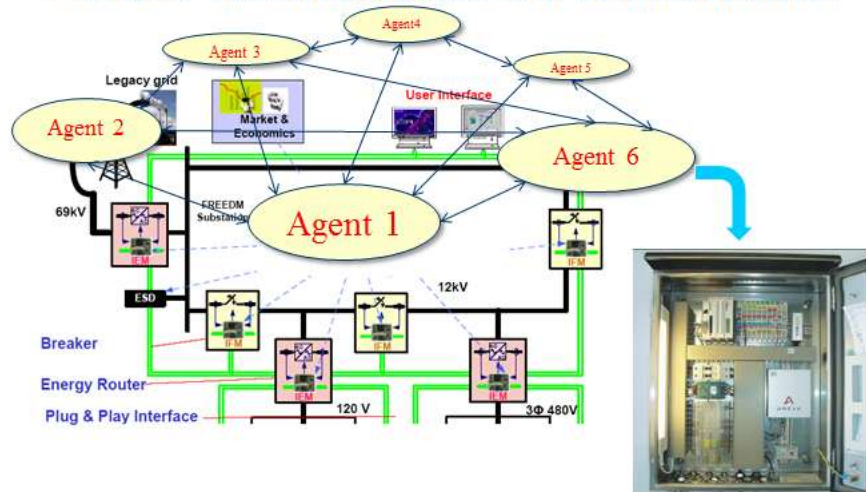
Die Hauptherausforderung des Projektes ist das SmartGrid Intelligence Engineering. Es ist klar, dass eine Erreichung der erklärten Ziele des Projektes mit starrer Automatisierung unmöglich ist.

Die Schlüsselfragen sind daher:

- Wie entwerfe und validiere ich verteilte Intelligenz?
- Wo und wie wende ich diese an?

intelligent Logical Nodes (iLN) of IEC 61850 for FREEDM

- Intelligent agents at the device level
- Implemented using new industrial standards
 - IEC 61850 – Object-oriented model for substation automation
 - IEC 61499 – Function blocks architecture for distributed automation



Die Kontrollarchitektur des FREEDM-Systems ist “multi-agent”. Die kommunizierenden Agenten versuchen, mehrere Funktionalitäten auf einmal zu erfüllen, von denen die wichtigsten Energiemanagement und Ausfallmanagement sind.

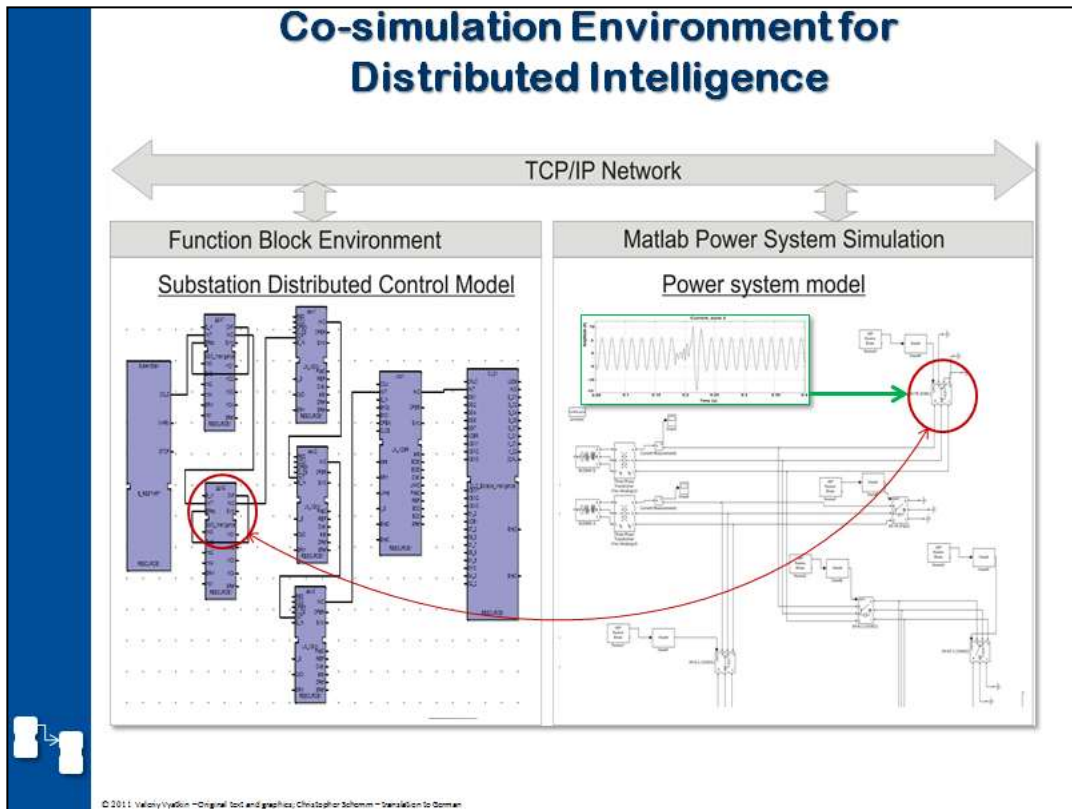
Das letzte Ziel ist es, die intelligenten Agenten auf Geräteebene einzusetzen, wie z.B. eine Schaltanlagensteuerung von AREVA, auf der Folie abgebildet.

Deshalb beschlossen wir, unsere iLN/61850-Architektur, basierend auf den neuen industriellen Standards, anzuwenden:

IEC 61850 – Object-oriented model for substation automation;

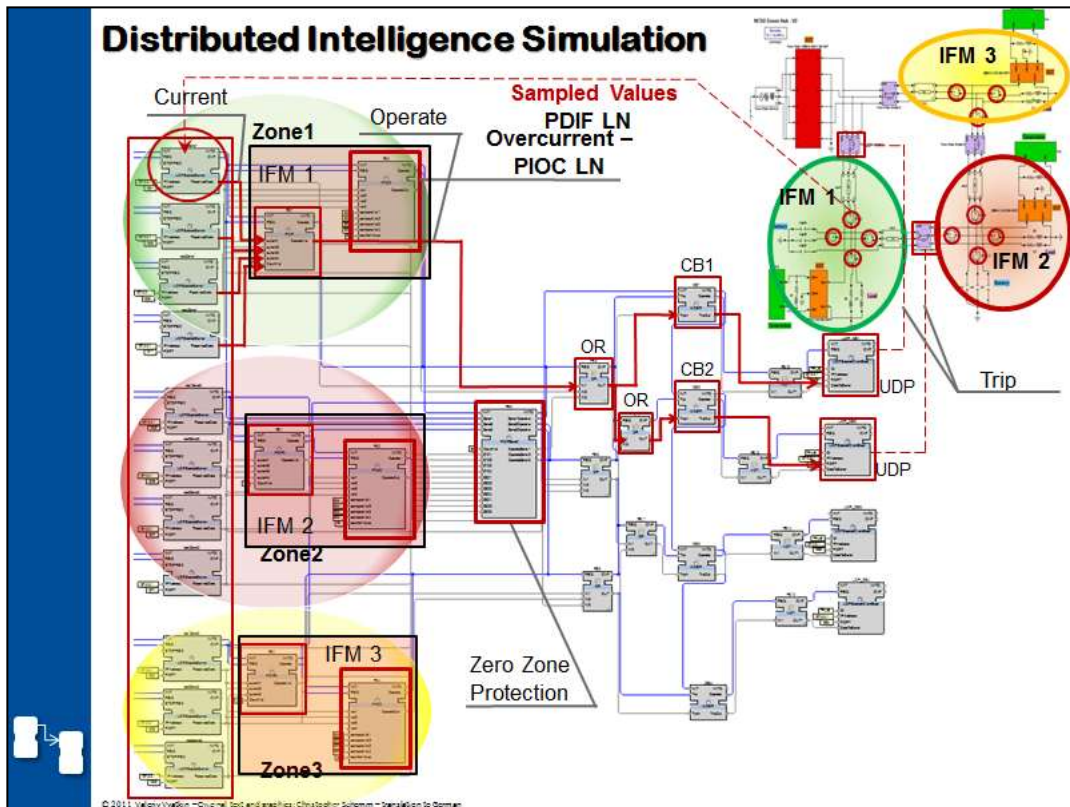
IEC 61499 – Function blocks architecture for distributed automation;

Der Entwurf einer solchen verteilten multi-agent-Software kann viel effizienter sein als traditionelle Ansätze. Dennoch stellt ihre Validierung eine weitere Herausforderung dar.



Um die verteilte Intelligenz zu simulieren, haben wir die Simulationsumgebung geschaffen, welche Matlab beinhaltet (mit “SimPowerSystem” package von Simulink) als ein Verhaltensmodell der realen Verteilungszweckmäßigkeit, und das verteilte Kontrollsystem der Zweckmäßigkeit in Funktionsbausteinen, parallel verlaufend mit dem Modell in einer Funktionsbaustein-Ausführung, wie in der Abbildung dargestellt. Die Kommunikation zwischen Matlab und der Funktionsbaustein-Umgebung wird realisiert durch die Verwendung von UDP-Schnittstellen.

Mit dieser Umgebung können wir genau die Dynamik physikalischer Prozesse modellieren und die Reaktion des verteilten Kontrollsystem sehen. Zum Beispiel können wir einen Fehler im Stromkreis “imitieren” und so die Überprüfung des Ausfallschutzes überprüfen.



Dies ist das FB-Kontrollmodell für das gesamte Fehlermanagementsystem. In der rechten oberen Ecke ist das Matlab-Modell des FREEDM Green Energy hub.

Es gibt drei Zonen im Kontrollmodell, das der Systemtopologie folgt. Jede Zone hat den gleichen Kontrollalgorithmus, daher sind die FBs die Instanzen der dazugehörigen Bausteine. Strom wird an den gegebenen Punkten gemessen und zum FB-Modell via UDP übertragen. Es gibt einen FB für jede UDP-Dose. Der geführte Strom wird zur differentialen Schutzeinheit geschickt: iPDIF – intelligente LN für differentialen Schutz. Jede Zone hat eine Instanz des iPDIF Funktionsbausteins.

Der differentiale Schutz bewertet die gegebenen Beispiele, und wenn er entscheidet, dass er den Fehler gefunden hat, sendet er das Operate (Trip)-Signal zum Stromkreisbrecher – CB – XCBR. Da sich in unserem Fall 2 Zonen CBs teilen, geht das Trip-Signal zuerst zum OR-FB. XCBR ist das Modell des CB im realen System.

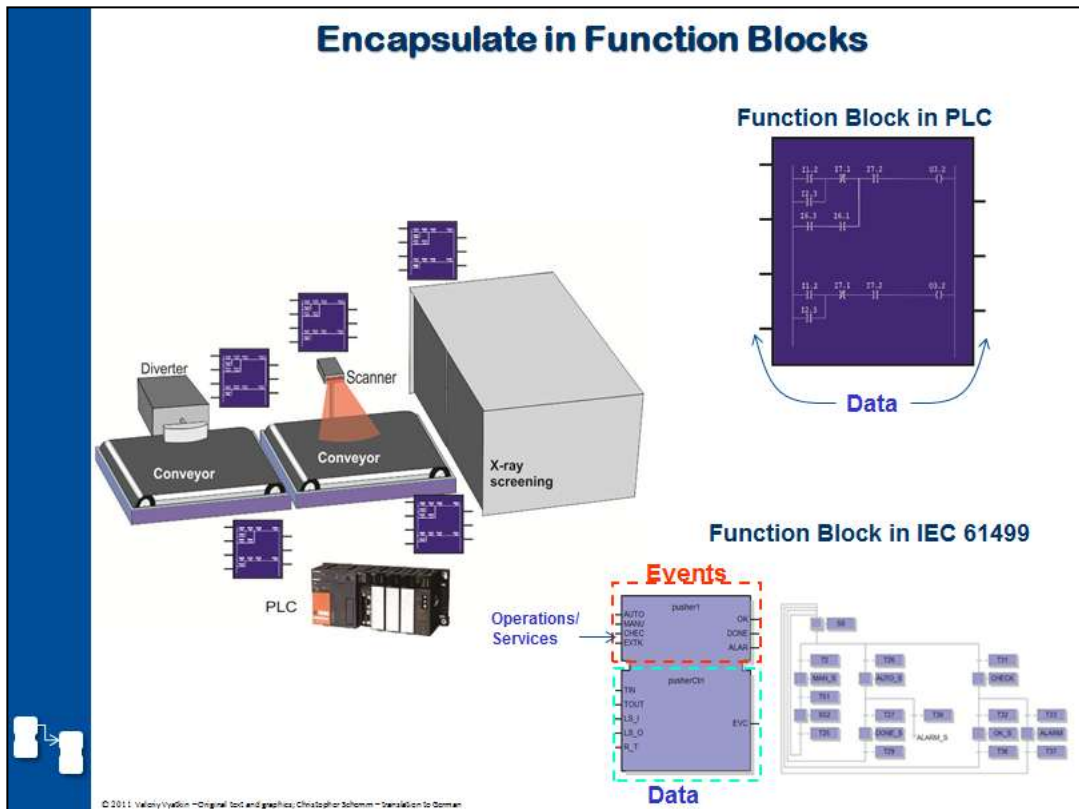
XCBR sendet das Trip-Signal durch UDP an das Simulink-Modell CB.

PIOC – ist Überspannungsschutz- sekundärer Schutz für dieses System.

Nullzonenschutz gibt es einen für das gesamte System – und es führt auch Back-Up-Schutz aus. Nullzonenschutz ist ebenfalls als differentialer Schutz implementiert – PDIF.

Jeder IFM bestehend aus PDIF und PIOC existiert in jeder Zone. Sie sind Agenten des gesamten Systems, treffen lokale entscheidungen und tragen zur verteilten Grid-Intelligenz bei.

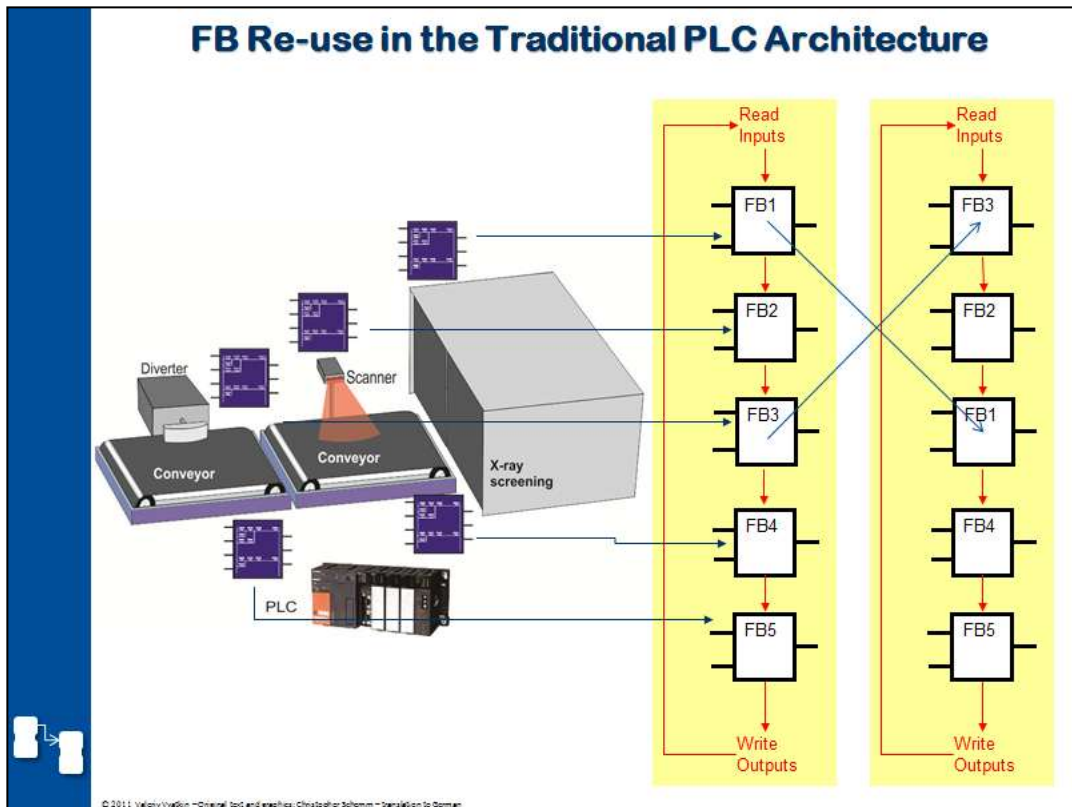
Wie Sie sehen können, ist jede funktionalität als Modul implementiert, es ist also sauber, einfach, dem System zu folgen, es zu warten, wiederzuverwenden neu zu konfigurieren.



Objektorientierter Entwurf mit SPS-Technologie ist wesentlich schwieriger. Angenommen, wir entscheiden uns, für jede mechanische Komponente des Systems einen Funktionsbaustein zu entwickeln, und anschließend ein SPS-Programm zu erstellen, welches diese Bausteine verwendet. Das würde verschiedene Probleme aufwerfen.

Es fehlt an interner Struktur in den Funktionsbausteinen des IEC 61131-3. Der gesamte Baustein hat nur einen Eingangspunkt, also wird es schwierig, ein effizient zusammenarbeitendes Verhalten von mehreren "Agenten" zu implementieren. Der FB-Code kann auf globale Variablen zugreifen, was den internen Code unsicher macht.

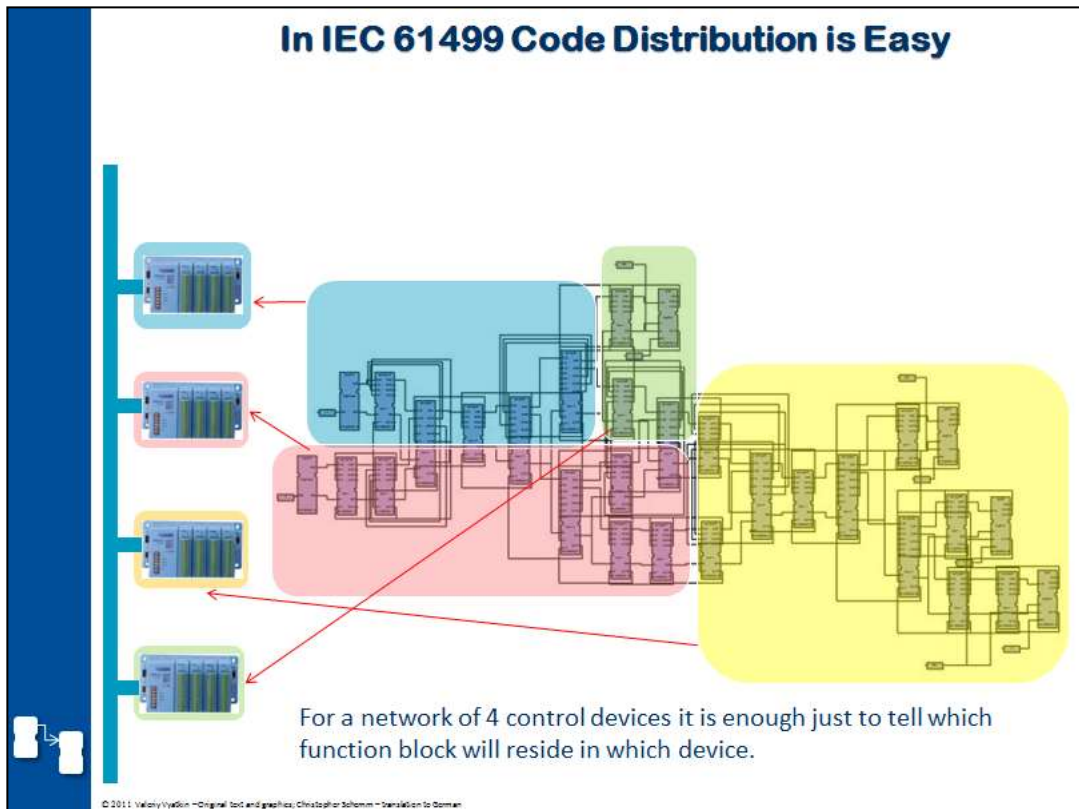
Alternativ sind funktionsbausteine der IEC 61499 den Objekten in der Objektorientierten Programmierung sehr ähnlich. Ihre Ereigniseingänge können als Methodenaufrufe interpretiert werden, und der Zustandsautomat kann dabei helfen, die Logik zu beschreiben, die verschiedene Dienste oder Operationsmodi kombiniert.



Im SPS-Scan werden alle Funktionsbausteine sequenziell während eines kurzen Zeitintervalls zwischen Update von Ein- und Ausgängen ausgeführt.

Dennoch ist die Abfolge dieser Aufrufe in keiner Weise durch die Beziehungen zwischen den physikalischen Geräten und Prozessen begründet.

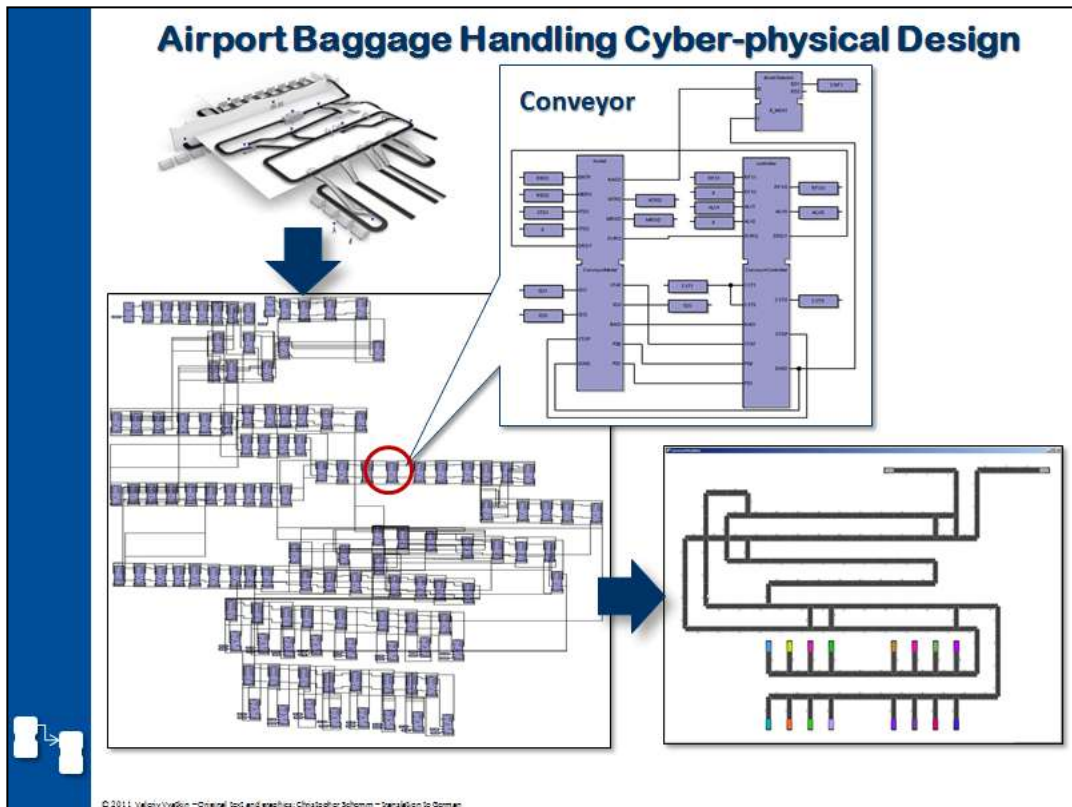
Als Ergebnis können unterschiedliche Entwickler eine unterschiedliche Ordnung festlegen, und das Programm würde sich anders verhalten.



Ein weiterer ernsthafter Vorteil des IEC 61499 ist die Leichtigkeit der Verteilung..

Zuerst wird die Funktionalität entwickelt als eine hardwareunabhängige Applikation, und erst im Anschluss muss eine bestimmte Hardware-Architektur ausgewählt werden.

Auf dieser Folie zum Beispiel sehen Sie verteilten Controllcode eines kleinen Flughafen-Gepäckhandhabungssystems. Für ein Netzwerk von 4 Kontrollgeräten ist es ausreichend, nur festzulegen, welcher Funktionsbaustein welchem Gerät innewohnt. Das Tool modifiziert automatisch den Code, um geräte- und netzwerkabhängige Kommunikationsfunktionen einzufügen, wo nötig.

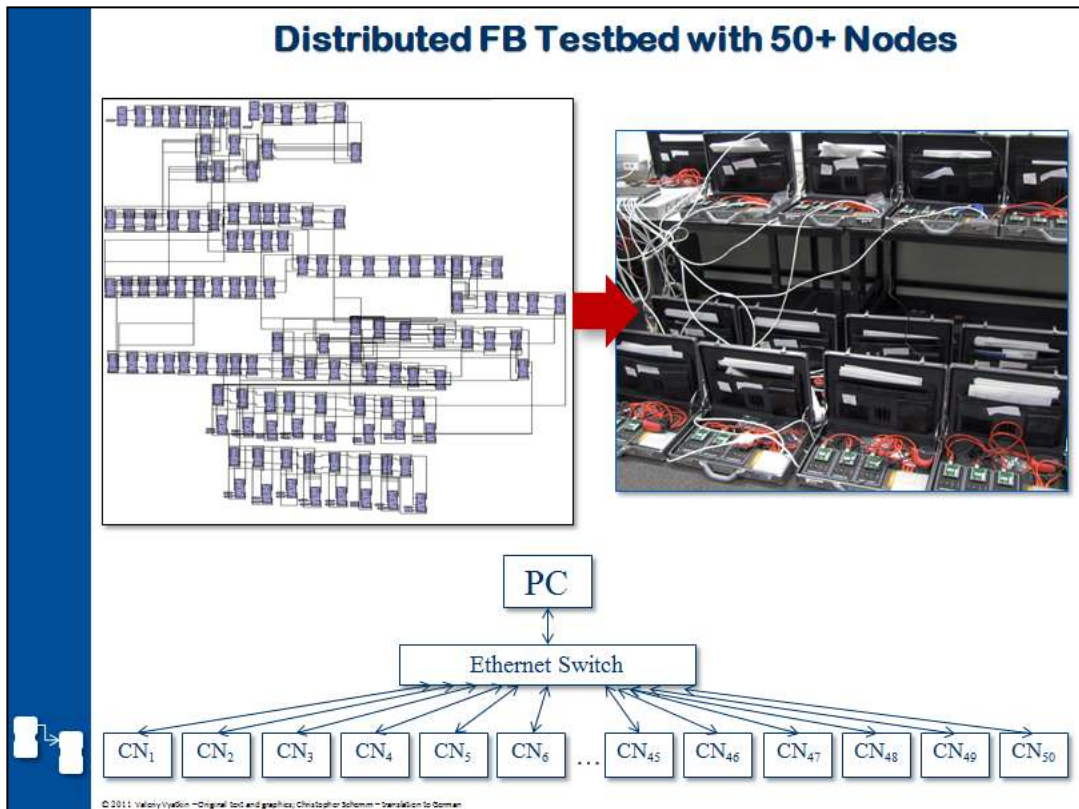


Das Flughafen-Gepäckhandhabungssystem (BHS) ist ein klassisches Beispiel eines komplexen Automatisierungssystems, welches Modularität und Verteilung mit hohen Performance-, Zuverlässigkeits- und Flexibilitätsanforderungen verbindet.

Vorhin haben wir das Konzept-BHS vorgeschlagen, bestehend aus *smart machines* mit eingebetteter Intelligenz. Solche Maschinen werden Sensoren, Aktoren und eingebettete Kontrollgeräte mit offener Software-Architektur integrieren, was einfache Plug-And-Play-Integrationen mit anderen solcher Maschinen zu komplexen, selbstkonfigurierbaren Systemen erlaubt. Jede Maschine wird durch einen intelligenten Agenten gesteuert – der Software, die traditionelle Low-Level-Kontrollfunktionen mit Kommunikation und Kollaboration mit anderen solchen Geräten kombiniert.

Basierend auf diesem Ansatz waren wir in der Lage, IEC 61499-Programme für die komplette Automatisierung eines kleinen Flughafen-BHS zu generieren. Es besteht aus den Dutzenden identischen Funktionsbausteinen – einer für jeden Beförderungsabschnitt. Jeder Funktionsbaustein ist intern nach dem MVC-Entwurfspattern strukturiert. Die generierte FB-Applikation kommuniziert auch mit automatisch generierten SCADA (siehe rechte Seite) . Als Ergebnis können wir:

- Die Operation am gesamten Flughafen simulieren;
- Den Code auf das gesamte verteilte Netzwerk von Controllern anwenden;



Um die Anwendbarkeit der Implementierung der verteilten Kontrolle, angewendet auf ein Netzwerk von eingebetteten Kontrollgeräten, zu zeigen, haben wir eine Testumgebung bestehend aus über 50 Kontroll-nodes an einem Netzwerk aufgebaut. Dies entspricht einem mittelgroßen Flughafen (oder einem autonomen Teil davon). Die Hardware ist heterogen und schließt Plattformen von vier Herstellern ein: WAGO, TCS-NZ, Netburner (alle ISaGRAF-konform) und Rockwell's Motor Drive Controller. Insbesondere beinhaltet die Testumgebung 48 Netburner-Geräte (der Kern des ISaGRAF demo-kits).

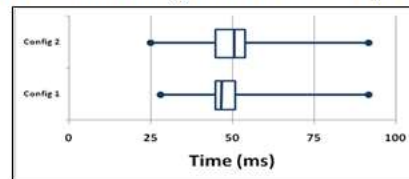
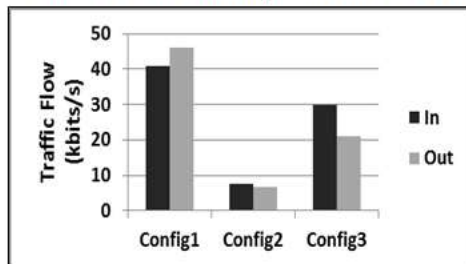
Diese Geräte repräsentieren ein prototypenhaftes Beispiel eines intelligenten, leichtgewichtigen Moduls, welches in jede Beförderung oder einen Motorenbetrieb integriert werden könnte.

Wir haben die Applikation durch die Benutzung von ISaGRAF workbench erfolgreich auf dieses Netzwerk angewendet. Dann haben wir mit verschiedenen Simulationsszenarien experimentiert. Es ist erwähnenswert, dass wir ziemlich genaue Dynamikmodelle von Beförderungsabschnitten in jeder Kontroll-Node betrieben haben.

Performance?



Network traffic and response time have been measured using PRTG Traffic Analyzer.



Event transmission delays between two control nodes

Network traffic for 3 selected control nodes

© 2011 Valery Vaidin - Original text and graphics; Christopher Schomm - Translation to German

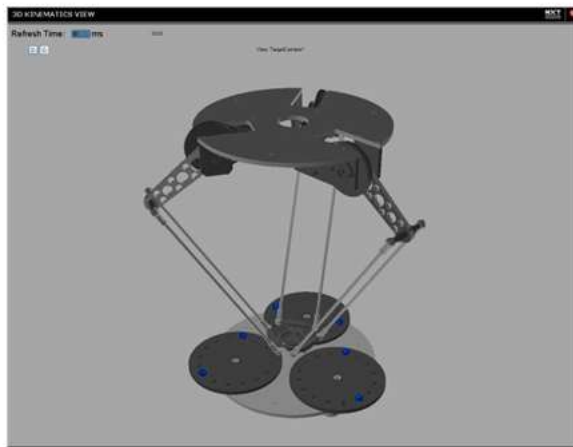
Um zu bestimmen, ob die Performance des Kommunikationsnetzwerks ausreichend ist für unseren verteilten BHS-Controller, wurden Analysen betrieben, um sicherzustellen, dass das ISaGRAF BHS-System die Bandbreite des Ethernets nicht überstrapaziert, und Tests durchgeführt, um die potenziellen Verzögerungen der Ereignisübertragung in einem belasteten System zu messen.

Ein verwalteter Switch wurde verwendet, um die Netzwerkauslastung zu messen. Verzögerungszeiten wurden durch die Gerätesystemzeit innerhalb der ISaGRAF Laufzeit gemessen. Diese Zeit wurde in Millisekunden gemessen, und Metriken wie derzeitige Durchlaufzeit, Gesamtanzahl der Durchläufe und Gesamtlaufzeit waren von Interesse. Drei Konfigurationen wurden analysiert. Config1 zum Beispiel betreibt 3 Beförderungs-FBs, wo alle von ihnen einen Umlenker auf einen Beförderer in einer weiteren Konfiguration beinhalten, und diese somit den höchsten Datenfluss hat.

Die momentane Implementierung des BHS-Systems in ISaGRAF ist weit davon entfernt, die Bandbreite auszureizen, die 100Mbit/s –Ethernet bietet. Mit Config1 zum Beispiel summiert sich die Benutzung auf 50KBit/s, was bedeutet, dass nur 0.05% der Ethernet-Bus-Bandbreite verwendet werden.

Die Verzögerung von Ereignisübertragung ist ein weiterer interessanter Parameter, da er dem Entwickler Einsicht in die Reaktionszeit des Systems gewährt, während das verteilte System unter einer Art Last arbeitet. Der Bereich der Verzögerung ist ziemlich bedeutend und variiert zwischen 25ms und fast 100ms für beide Konfigurationen. Dennoch schlagen die meisten Ergebnisse eine Verzögerung der Ereignisübertragung von ungefähr 50ms vor.

Delta Robot



- Fully implemented in IEC 61499
- Distributed system with 3 control nodes: SIEMENS, Beckhoff, Wago + SICK safety controller
- Motion control with two VFDs on EtherCAT
- Complete 3D simulation in NxtControl



© 2011 Valery Vvedin - Original: toC and graphics: Christopher Schlemm - Translation to German

Um die IEC 61499 der Performance verlangenden Motion-Control-Anwendung zu demonstrieren, hat NxtControl einen Delta-Roboter entwickelt, mit Steuerung, die voll mit IEC 61499 implementiert ist. Die Kontrollarchitektur schließt ein verteiltes System von 3 Kontroll-Nodes ein: SIEMENS EC 31, Beckhoff CX 1010, Wago Ethernet coupler und SICK safety controller.

Die Bewegungssteuerung wurde durch die Verwendung zweier VFDs auf EtherCAT Implementiert. Zudem wurde eine komplette 3D-simulations des Roboters mit NxtStudio entwickelt. Der Roboter wurde auf der SPS/IPC/Drives-Messe im Jahr 2010 ausgestellt.

Dieses Roboter haben wir jetzt in unsere Lab in Auckland!

Conclusion

Function block architecture of IEC 61499 combines all essential features of a mature model-based software engineering framework with:

- System-level design of distributed systems
- Distribution on networked targets
- Open standard
- Determinism and efficiency of execution



© 2011 Valery Vyshin - Original text and graphics; Christopher Schmitt - Translation to German

Das Begreifen des IEC 61499 hat ziemlich lange gedauert, da diese Technologie neu ist im Gegensatz zum vorherigen IEC 61131-3-Standard, welcher eine Anerkennung von vergangenen Praktiken war. Dennoch sicherte der ausgiebige Forschungsaufwand in den letzten zehn Jahren das sehr solide Gepäck von Ergebnissen, das von der Industrie genommen und verwendet wird. Nun können erste Ergebnisse auf dem Markt beobachtet werden. .

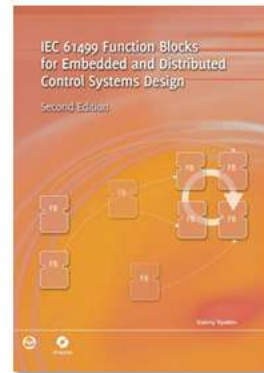
Mit den ersten kommerziellen, konformen Tools und Geräten können Benutzer bereits die Vielzahl an “Entwurfszeitvorteilen” von IEC 61499 Toolketten für Entwurf auf Systemebene und Implementierung von Automatisierungssystemen genießen. Andere Vorteile von IEC 61499, wie Portierbarkeit und Interoperabilität, waren bisher keine große Sorge der Hersteller. Dennoch sind diese Features “eingebettet” und werden hoffentlich in nächster Zeit bemerkt werden.

Die weitere Adaption von IEC61499 wird der Industrie helfen, vom Versprechen der Forschungsergebnisse in Sachen computerintegrierter und intelligenter Automatisierung zu profitieren , welche die Plattform bieten für die Anwendung von verteilten Applikationen mit Dutzenden oder sogar Tausenden von kommunizierenden, autonomen Kontroll-Nodes.

Weitere Information

Buch:

Vyatkin, V., IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design, 297 pp., Instrumentation Society of America, USA, July, 2007 – Second Edition in print
To order: <http://www.isa.org/iec61499>



Artikel:

V. Vyatkin, "IEC 61499 as Enabler of Distributed and Intelligent Automation: State of the Art Review", IEEE Transactions on Industrial Informatics, 2011, in print

Check web-site: <http://www.ece.auckland.ac.nz/~vyatkin/>



© 2011 Valery Vyatkin - Original text and graphics; Christopher Solomon - Translation to German

Buch:

Vyatkin, V., IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design, 297 pp., Instrumentation Society of America, USA, July, 2007 – Second Edition.

Zu bestellen: <http://www.isa.org/iec61499>

Artikel:

V. Vyatkin, "IEC 61499 as Enabler of Distributed and Intelligent Automation: State of the Art Review", IEEE Transactions on Industrial Informatics, 2011, in print

Check web-site: <http://www.ece.auckland.ac.nz/~vyatkin/>

IEC 61499 Day at SPS/IPC/Drives

23rd of November 2011, Nuremberg, Germany

9:30 – 16:00, open doors policy

Organizer: Prof Valeriy Vyatkin, University of Auckland, New Zealand

Sponsors: ISaGRAF and NxtControl

Program:

1. Workshop “Industrial experiences of IEC 61499 application”;
2. Lunch-time presentations by leading researchers;
3. Hands-on trainings on ISaGRAF and NxtControl tools;

The workshop will reflect industrial experiences, in particular of Tessmar (Austria), ITIA-CNR (Italy), Glidepath (New Zealand), PROFACTOR (Austria), Visual Components Oy (Finland), Tuomas Group (Spain) and Energex (Australia) in applying IEC 61499 technology provided by software tools of ISaGRAF(Canada) and NxtControl (Austria) and using off-the-shelf hardware of Advantech, Beck, Beckhoff, FESTO, SIEMENS, Wago and other suppliers.

Admission: Free for SPS/IPC/Drives visitors and exhibitors

To register contact: iec61499day@gmail.com

More details: <http://www.ece.auckland.ac.nz/~vyatkin/iec61499/day.pdf>



© 2011 Valeriy Vyatkin - Original text and graphics; Christopher Schmitt - translation to German