4DIAC

A Framework for Distributed Industrial Automation and Control

Alois Zoitl

fortiss GmbH
An-Institut Technische Universität München
Outline

- Background Industrial Automation
- Short introduction to IEC 61499
- Application example
- Overview 4DIAC
- Evaluation examples
Background Industrial Automation

- Industrial plants are most of the time *one of a kind*
- Low programming skills of typical industrial plant maintenance personnel
- General hardware platforms → Programmable Logic Controller (PLC)
  - Modular extensible architecture
    → re-usable in many different systems
  - Simplified real-time programming through *dedicated languages* and *defined execution model*
- New demands
  - Adaptability
  - Shorter product life-cycles, global markets
    → *Industry 4.0*
- Requires for new control architectures and programming methods
IEC 61499 Introduction

• Origin
  – 1990s: holonic and agile manufacturing systems
  – Requirements: flexibility, adaptivity, and distribution

• Goals
  – Standardized architecture for function blocks in distributed industrial-process measurement and control systems
  – Logical equal control devices
  – Not necessarily a supervisory control device
  – Basic support for dynamic reconfiguration

• Developed by IEC TC65/WG6
  – Started 1993
  – Experts originally from USA, Germany, Japan, UK, Sweden, France, Italy
  – Also responsible for
    • IEC 61131-3 - Programmable Controller Languages
    • IEC 61131-8 - Programmable Controller Language Guidelines
Core Element: Function Block

- Function Blocks extended with event interface
- Pure **event-driven** execution model
- Data types based on IEC 61131-3
- Focus on **encapsulation** and **reuse**
- No global or directly addressed variables
- Hardware access with special function block type: **Service Interface Function Block**
IEC 61499 Application Model

- Function Blocks
- Event connections
- Data connections

Diagram:

- FB 1
- FB 2
- FB 3
- FB 4
- FB 5
System and Distribution Model

System Model:
- Devices
- Process/Machine
- Communication Infrastructure

Application Model:
- FB 1
- FB 2
- FB 3
- FB 4
- FB 5

Controlled Process/Machine:
- Device 1
- Device 2
- Device 3
- Device 4
- Device 5

Applications:
- Application 1
- Application 2
- Application 3
Control Application Example

- Fill level control of a tank
- Pump draining independently → disturbance
- Simulated tank, valve, and pump
- Simple HMI
  - Set-point
  - Hand / automatic
  - Strip chart

Based on an Example from James Christensen
Central Application

Test in Simulation Device
Model System
Distribute to Devices
Device Specific Parameters and Adjustments
Deployment
Overview 4DIAC

Framework for Distributed Industrial Automation and Control
4DIAC Overview

- **Goal:** open source implementation of IEC 61499
  - Common basis for development, industrial application, and research of IEC 61499
  - Leveraging the use of IEC 61499 within industry

- **Provided Content**
  - **4DIAC-IDE:** engineering tool
  - **4DIAC-RTE (FORTE):** small real-time capable runtime environment
  - **4DIAC-LIB:** function block library
  - **4DIAC-Systems:** example applications

- **Open Source License**
  - Eclipse Public License
  - Allows usage in products and proprietary add-ons
4DIAC Users

- **Industrial users**
  - nxtControl: FORTE with own tool
  - Bachmann: evaluation
  - Voigt + Wipp Engineers: evaluation
  - ABB: use in internal research projects

- **Universities and research institutes**
  - Austria, Germany, Spain, Italy, New Zealand, Canada, …

- **Application domains:**
  - Building automation, process industries, laboratory automation, smart grids, machine control, sequence coordination, …
4DIAC-IDE: Application/Distributed System Development

- System Editor:
  - Application modeling
  - Support for sub-applications
  - Device and network specification
  - Mapping
- Deployment:
  - Support for different profiles
  - Separate deployment possible
- Project Management
  - Project specific Type-Libraries
4DIAC-IDE: Function Block Development

- **Types**
  - Basic FB’s
  - Composite FB’s
  - Service Interface FB’s
  - Adapters

- Code generation (C++)
4DIAC-IDE: Testing, Monitoring, Debugging

- Test FBs
  - On target device
  - Manual
  - Automated unit tests
- Investigate Applications
  - Watch interface elements
  - Trigger events
  - Force values
FORTE: Overview

- Written in C++
- Modular design → scalability
- Core functionality
  - Basic IEC 61499 elements and their execution
  - Hardware abstraction
- Most extensions in FB libraries
- Real-time execution support
  - Requires real-time operating system
- Flexible communication infrastructure via so called communication layers.
- Currently Supported layers are:
  - FBDK ASN.1 encoding
  - Ethernet
  - Modbus TCP client
  - OPC DA client
FORTE: Supported Platforms

- **Operating Systems**
  - Windows
  - Posix: Cygwin, Linux (i386, PPC)
  - NET+OS® 7
  - eCos

- **Devices and Boards**
  - Digi Connect ME® (ARM7)
  - Lego Mindstorms nxt (ARM7)
  - Bachmann electronic M1 PLC
  - KIPR’s CBC v2 robot controller

- **Further tested and upcoming**
  - e.g., Raspberry Pi
4DIAC Toolchain

Application download

FORTE firmware type file generation

Target Compiler

Platform specific FORTE version

FORTE source files

.h

.cpp
Evaluation Examples

- Bin Picking Control
  - Integration of 3D vision algorithms
  - Automatic collision free path planning
  - Interaction with robots

- Closed Loop Control
  - Real-time constrained execution
  - Reconfiguration during full system operation
  - Networked closed-loop control

- Discrete Manufacturing
  - Coordination control of several different machines
  - Up to 120 control devices

- Smart Grid and Laboratory Automation
  - Integration with IEC 61850
  - Interaction with Distribution Management Systems and SCADA
Contact

Dr. Alois Zoitl
fortiss GmbH
Guerickestraße 25, 80805 München, Germany
Tel +49 89 3603522 22
Fax +49 89 3603522 50
alois.zoitl@fortiss.org
www.fortiss.org

Acknowledgements
• James Christensen
• Gerhard Ebenhofer
• Ingo Hegny
• Carolyn Oates
• Thomas Strasser
• Christoph Sünder