Service-Oriented Architecture in Industrial Automation

Apply SOA to IEC 61499 Function Blocks

William Dai  Postdoc SRT-DCC
Introduction

Background

Motivation

IEC 61131-3 PLCs and IEC 61499 Function Blocks

Service-Oriented Architecture Principals in Automation

Apply Service-Oriented Architecture in IEC 61499

Case Study

Development Plan
Background

**Study**

2003 – 2006  BE(Hons)  in Computer Systems Engineering  
University of Auckland, New Zealand

August 2008 – July 2012 PhD in Electrical and Electronic Engineering  
University of Auckland, New Zealand

**Academic Work**

February – May 2013  
Part-Time Lecturer  
UNITEC Institute of Technology, New Zealand

March – June 2013  
Professional Teaching Fellow  
University of Auckland, New Zealand

**Publications:**  
3 IEEE Transaction Papers(TASE, TII, TSMC)  
Several IEEE Conference Papers
Background

**Industry Work**

Jan 2007 – June 2010  Software Engineer
Glidepath Group, New Zealand

July 2010 – Current Sub-Contractor of
Glidepath Group, New Zealand

**Completed Projects**

More than 60 airport baggage handling systems and parcel sortation systems projects worth more than 100M USD
(Nassau, Perth, Male, Liberia, Auckland, Christchurch, Durban, Muscat, Toronto, Guangzhou, Penang, Australia Post...)

**Personal Project**

PLCLink – PLC Testing App Available on App Store

**R&D Projects**

(PLC Code Generator, BHS Emulator, Ethernet/IP Remote IO Simulator, CIP and Ethernet/IP Interface, Web based Reporting system, Common Serial Interface, Documentation Generator)
Motivations

Lack of Interoperability and Flexibility in PLCs

IEC 61131-3 Standard not for Distributed Automation Programming

IEC 61499 standard is not the main stream in Industrial Automation

Current implementation of IEC 61499 standard is also not "smart"
IEC 61131-3 PLCs

Scan Cycle Based (Event Trigger could be achieved)

Support Variable Types: Input, Output, Local, InOut (FB Only)

Support Nested FB Structure

Support Four Languages + SFC

Instance Support – Function Blocks Only
Scan Cycle Based Execution (Synchronous)

- **Read**
  - Read Inputs
  - Each scan go through all tasks defined in the PLC
  - Each task can have one or more programs
  - Each program can be written in any IEC 61131-3 languages

- **Process**

- **Write**
  - Update Outputs
Scan Cycle Based Execution (Asynchronous)

- Read
- Process
- Write
How does it look like?
IEC 61131-3 PLC Issues

Compliance with IEC 61131-3 but not compatible with each other's implementations.

Each PLC brand has their own implementations of IEC 61131-3.

4 PLC Programming Languages + SFC defined in IEC 61131-3 but may not always be available in every PLC platform.
Event Trigger

Support Variable Types: Input, Output, Local (BFB and SIFB)

Support Nested FB Structure (CFB)

Support all IEC 61131-3 Languages and more

**IEC 61499 Function Blocks**
IEC 61499 FB Introduction

Three Function Block Types
(Basic, Composite and Service Interface)

Basic FB is controlled by a state machine called Execution Control Chart (ECC)

Composite FB consist of a network of FBs

Service Interface FB is a “Block Box”

System Configurations are the top level structure in IEC 61499 contains FB applications, resources and devices
CASE STUDY IN BHS
HOW COMPLEX BHS COULD BE
CASE STUDY
BHS SCREENING SUBSYSTEM
Distribute by Layout
IEC 61131-3 VS IEC 61499

**IEC 61131-3 (4 Steps):**
- Copy Definitions
- Redefine I/O Ownership
- Move Instances
- Create Communication

**IEC 61499 (2 Steps):**
- Change Network Configuration
- Change Deployment Mapping
Service-Oriented Architecture Concept
## WHY SOA IN IEC 61499 NOT IEC 61131-3

<table>
<thead>
<tr>
<th>SOA PRINCIPLES</th>
<th>IEC 61131-3</th>
<th>IEC 61499</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reusability</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Service Contract</td>
<td>No</td>
<td>Partially</td>
</tr>
<tr>
<td>Loose Coupling</td>
<td>No</td>
<td>Partially</td>
</tr>
<tr>
<td>Abstraction</td>
<td>Partially</td>
<td>Yes</td>
</tr>
<tr>
<td>Composability</td>
<td>Partially</td>
<td>Partially</td>
</tr>
<tr>
<td>Autonomy</td>
<td>Partially</td>
<td>Partially</td>
</tr>
<tr>
<td>Statelessness</td>
<td>Possible</td>
<td>Possible</td>
</tr>
<tr>
<td>Discoverability</td>
<td>No</td>
<td>Partially</td>
</tr>
</tbody>
</table>
BENEFITS OF APPLYING SOA IN PLC
FLEXIBILITY – AUTOMATIC DEPLOYMENT
BENEFITS OF APPLYING SOA IN PLC

FLEXIBILITY

PLUG & PLAY (AUTOMATIC LOADSHARE)
BENEFITS OF APPLYING SOA IN PLC
FLEXIBILITY
AUTOMATIC FAULT DETECTION AND RECOVERY
BENEFITS OF APPLYING SOA IN PLC
INTEROPERABILITY
RESOLVE COMPATIBILITY ISSUE
BENEFITS OF APPLYING SOA IN PLC

INTEROPERABILITY

USE EXTERNAL SERVICE LIBRARY
**APPLY SOA IN IEC 61499**

**BASIC PRINCIPAL**

Event used as Message Type Name
Data variables are stored in Message Contents
Possibly return data variable value back to Service A

Service Repository in Each PAC

Service A → Register
Search/Result

Service B → Register

Service Repository in Each PAC
APPLY SOA IN IEC 61499
APPLY SOA IN IEC 61499 BHS EXAMPLE
Development Plan

SOA FB IDE

SOA PLC Runtime

SOA Integrated SCADA/HMI
What makes ours different from others

IDE

Usability
Service Store - “Apps”

Runtime

Downtimeless
Intelligent

SCADA/HMI

Easy to Build (Integrated in Design)
Easy to Access (service calls)
Questions

William Dai
William.dai@ltu.se