25

A LOW-COST EXPERIMENTAL SYSTEM AND ENGINEERING METHODOLOGY FOR IEC 61499 APPLICATIONS

Hirotsugu Tsunematsu[1], Hisashi Sasajima[1], Tatsuya Hojo[2]
[1] *Yamatake Corporation, {tunematu, sasajima }@atc.yamatake.co.jp*
[2] *Yamatake Corporation, hojo@atc.yamatake.co.jp*

IEC 61499, a new concept of control platform that consists of intelligent devices, has been proposed to resolve complications in system development and modification. However, there are few examples that have applied this architecture to a real system and the design methodology has not been established yet. This paper proposes a low-cost experimental system using LEGO® blocks for a feasibility study of design methodology based on IEC 61499 concepts. A method of system construction and technique to adopt features of physical machine factor, which was studied through using this experimental system, will also be presented.

1. INTRODUCTION

In general, conventional machine control systems consisting of centralized systems with PLC or distributed systems with some PLCs have been used. However, developing control software of these systems is very complicated, and system developers have sought a new control platform that can be configured and modified with ease.

To meet such requirements, a new concept of control platform with distributed intelligent devices has been proposed. It consists of intelligent devices such as axis controls or conveyance controls. Instead of a control program implemented in a centralized program, the control functions of each device are implemented in its own control unit as function blocks. Therefore, know-how of controlling devices can be embedded into each device. These devices will be easy to use, and the system developer doesn't have to open their knowledge. The system integrator can develop the system software in a cost-effective method of combining the distributed functions in the device. In addition, the software of these devices becomes highly reusable because it is isolated from the functions of other devices.

The International Electrotechnical Commission (IEC) is working to standardize this architecture as IEC 61499. However, only a few examples of applying this architecture to an actual system exist, and the design methodology has yet to be established. Also many of examples were shown with a simulation or discussed on conceptual design only.

V. Mařík et al. (eds.), *Knowledge and Technology Integration in Production and Services*

In this paper, we propose a low-cost, experimental system using LEGO® blocks for the study of design methodology based on IEC 61499 architectures. This system consists of several sensors, motors and gears, and is controlled with two compact PCs. The devices are controlled with function blocks, which are compliant with IEC 61499. Because the controlled target is built with LEGO® blocks, the system can be developed and modified with ease. It also fits the IEC 61499 concept of building a whole system by putting parts, i.e., function blocks together like LEGO® blocks. We will also present a method of system construction and a technique to adopt features of physical machine factor, which was studied through using this experimental system.

2. BASIS OF IEC 61499

2.1 System Model

A system model of IEC 61499 concepts is shown in Figure 1. The controlled process consists of multiple devices. Each device is connected to a network and is able to communicate with each other. It also has its own functionality and control applications for this process, which are realized using functions on one device or combining functions distributed on multiple devices.

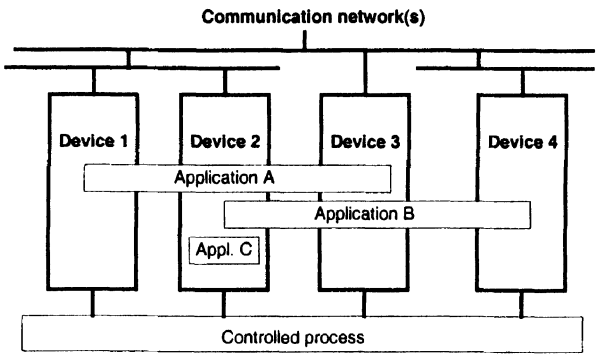

Figure 1 – System model

2.2 Application Model

Figure 2 illustrates an abstractive model of a distributed application. The application consists of one or more function block instances connected by event and data connections.

The function block instances may be distributed among devices. Execution sequence is decided by event flow, and data transfer among function blocks is represented by data flow.

In an executable model, event and data connections among devices can be represented by Service Interface function blocks.

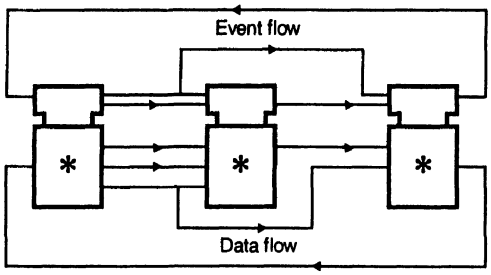

Figure 2 – Application model

2.3 Execution Control Chart

Execution of algorithm in Basic Function Block, which cannot be decomposed into other function blocks, is controlled by Execution Control Function which is represented by Execution Control Chart (ECC) as shown in Figure 3 (b).

In this example, *EC initial state*, which is active upon initialization of ECC, is START. In response to an INIT event input, *EC state* will be transferred to INIT state where *EC action* that initializes the function block INTEGRAL_REAL will be executed. First, the corresponding INIT algorithm is executed, and upon completion of execution, event output INITO will be generated. Condition 1 then sets the INIT state back to START state. Similarly in response to an EX event input, EC state is transferred to MAIN state followed by the execution of EC action, after which condition 1 sets the MAIN state to START state.

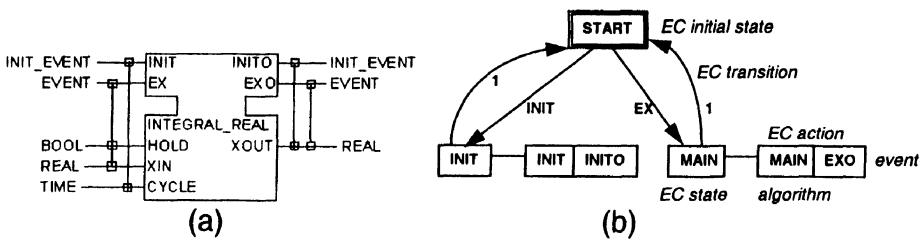

Figure 3 – Execution Control Chart

2.4 Composite Function Block

To avoid complexity of an application, the concept of *Composite Function Block* has been defined. In Composite Function Block, algorithms and their execution control are specified through event and data connections in one or more function block networks. It not only reduces the complexity of applications consisting of multiple function blocks but also encapsulates some specific know-how of control algorithms.

3. EXPERIMENTAL SYSTEM

3.1 Controlled Object

As illustrated in Figure 4, the controlled object was assembled of LEGO® blocks to assess IEC 61499 concept design methodology. The function of this machine is to feed and carry objects like an autonomous warehouse. It consists of three parts: one feeder and two conveyers. Each part is aimed to be an intelligent device and has sensors and a motor. These feeder and conveyer devices are controlled with low-cost AT-compatible computers running Windows 98/NT with PC 104-based digital input

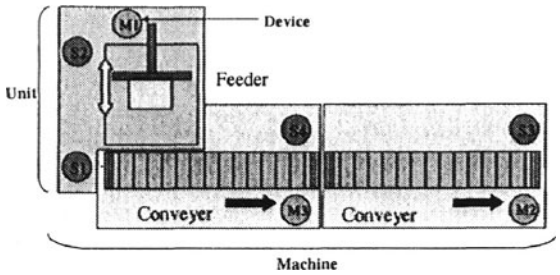

Figure 4 – Experimental equipment

and output control boards.

Supposing these devices are intelligent devices, each device should have an individual controller, however two conveyers are controlled by the same controller due to a restriction of the equipment.

3.2 System Overview

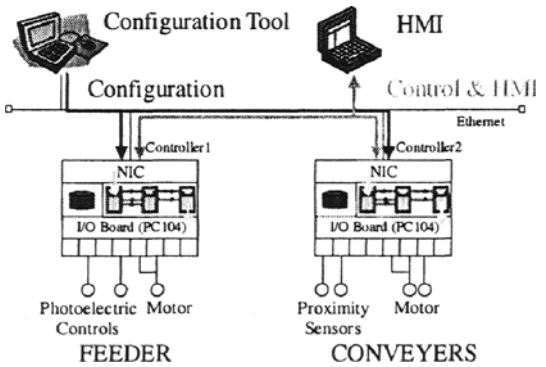

Figure 5 – System Overview

A system overview is shown in Figure 5. One PC is used for configuration of the devices and the other is used for operation.

In regards to configuration, each device has a function block library. According to the configuration message sent from the configuration tool, the function block instances and event/data connections will be created.

This configuration tool is not necessary for control and monitoring. Local applications on devices and HMI on remote PCs will communicate with each other through communication interface function blocks. Sensors and motors are controlled through process interface function blocks.

3.3 Operation

After a carried object has been set at designated place, the operator will press the "START" button on the HMI. This button will trigger a message sent to the "FEEDER" device, and the FEEDER will then push out the object onto the first CONVEYER. When the FEEDER accomplishes its work, a message will be passed to the first CONVEYER. In the same way, the first CONVEYER will take action to carry the object and send a message to the second CONVEYER. After the object arrives at the end of the second CONVEYER, a message will be forwarded to the first CONVEYOR and both conveyers will stop.

4. APPLICATION DEVELOPMENT METHODOLOGY

4.1 Overview

System development can be separated into several layers of physical interface, devices, units, and applications. In each layer, components supplied by the lower layer and some layer's specific know-how given by a developer will be encapsulated into a new composite function block.

This section describes the steps of feeder unit control design of the experimental system and the development of the overall application.

4.2 Physical Interface Layer

Introduction of service interface function blocks provides an application that is a useful interface in the physical world. In this experimental system, the network communication function block is furnished with the Function Block Development Kit (FBDK), an IEC 61499-compliant development environment provided by Rockwell Automation as a communication interface. This section discusses the process interface function block for digital input and output.

The controller has a digital input/output board with eight inputs and eight outputs. A driver of the interface board is installed on the OS, thus the function block is designed to encapsulate this driver. Due to restrictions of the driver, digital input and output interface have to be implemented into the same function block as Digital Input/Output interface.

Figure 6 describes the service sequences of this function block.

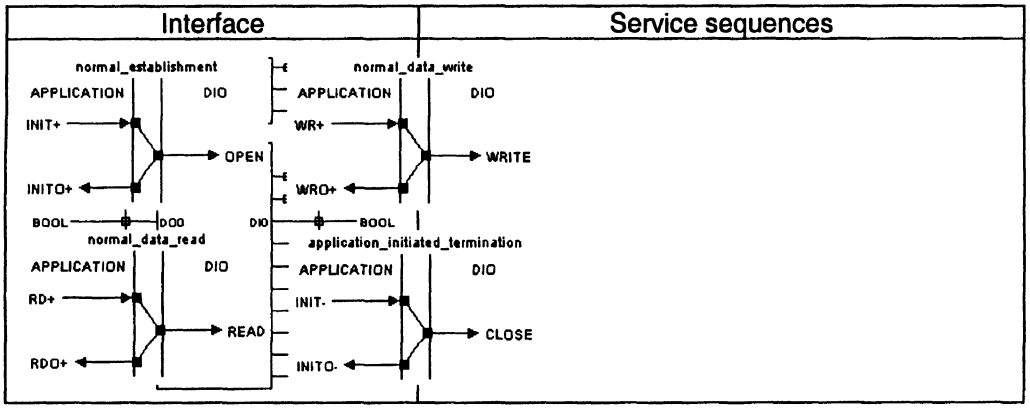

Figure 6– Digital Input/Output interface function block

4.3 Device Layer

4.3.1 Motor Control

A motor is a component device of the unit. Motor control function block implements know-how of the motor control designer.

A motor needs to move forward and backward, and stop at an exact location. As shown in Figure 7, this function block has M1_FW, M1_BW and M1_STOP event inputs. Each event initiates state transition to move or stop. In the M1_FW state, the M1_FW algorithm is executed in order to turn on the digital output port that specifies the motor direction. Then the REQ event is generated.

The "STOP" function includes a special sequence, which is a part of the know-how of motor control. When a motor stops, it should be reversed for a short time to brake. Duration of reverse depends on physical factors, such as mass. Therefore, this function block should be connected to the event delay function block that decides the reverse duration. The "STOP" event output is connected to the "START" event input of E_DELAY. After the duration of reverse, the E_DELAY generates "EO" event and will affect the state transition of the motor control function block to turn off the motor. Above two function blocks should be a composite function block.

4.3.2 Sensor Interface

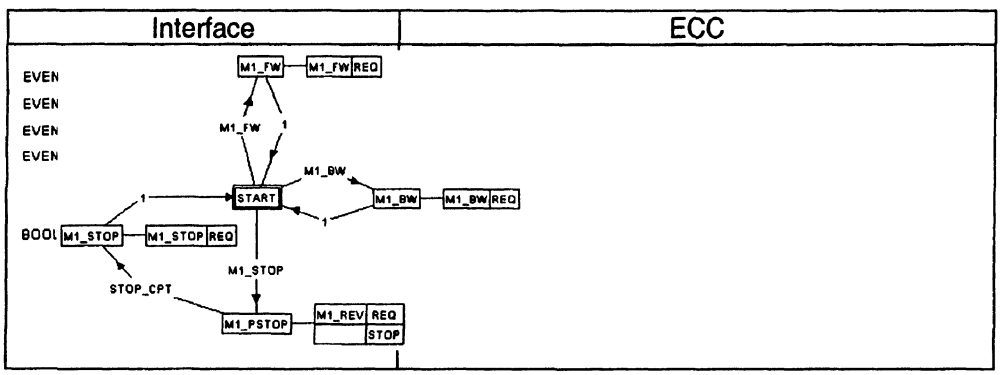

Figure 7 – Motor Control function block

For a machine control designer, state transition is the only necessary information with regards to sensors. Therefore, the sequence of reading the sensor cyclically, comparing the sensor's state and notifying the sensor's state transitions, is enclosed into the composite function block. In order to inspect sensor state change, the "MASK" function block is used. This function block compares the last and current states, and only when there are differences, it will output "IND" event.

4.3.3 Motor-with-Sensor Device

The above two devices can be composed into a function block of intelligent devices as shown in Figure 8. Know-how of using motor and sensor is encapsulated into a function block and it is a highly reusable software module.

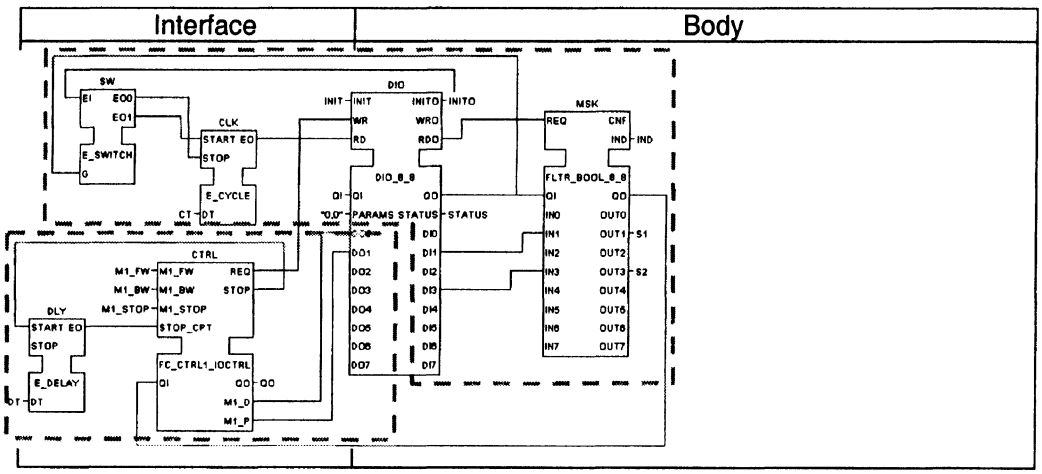

Figure 8 – Motor-with-Sensor function block

4.4 Unit Layer

There are four units in this experimental machine, i.e., FEEDER unit, HMI unit and two CONVEYER units.

Functions of the feeder unit can be implemented as the feeder control function block. This control function block is connected to a motor-with-sensor control function block. The combination of feeder control and motor-with-sensor control can be a feeder composite function block, FEEDER.

Considering this unit will be used under a distributed system, the local application in the feeder unit should provide the interface to get initiation of the feeder sequence and send a notification message confirming the sequence.

HMI unit and CONVEYER units can be built similarly.

4.5 Application Layer

Applications can be made combining HMI, FEEDER, and CONVEYER. Since every unit has a local application designed with function blocks and the local

application has network interfaces, these devices can be combined easily through network.

5. CONCLUSION

An IEC 61499-compliant system design methodology using low-cost experiment system built with LOGO® blocks was discussed.

As shown in Figure 9, this methodology of developing machine control application will provide the hierarchy of system development of H/W, S/W suppliers, device vendors and system integrators based on IEC 61499 technologies. Control know-how of components can be hidden in function blocks, while it is available for upper layer users. As a result of know-how encapsulation repetition, the system integrator will be able to accomplish complicated system development with ease.

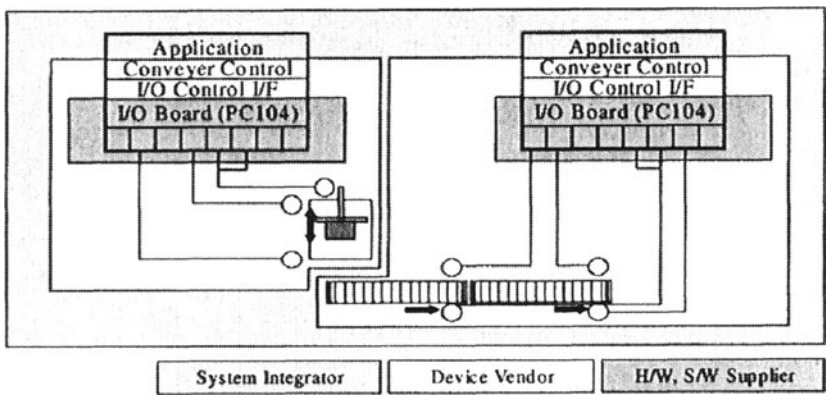

Figure 9 – Future System Development

For further research of IEC 61499 capabilities, we will study more complex and realistic examples including the case of system enhancement. Also, we will compare the scalability and development efficiency of IEC 61499 with conventional PLC programming or alternative implementation in C++, etc.

6. ACKNOWLEDGEMENTS

This work was accomplished using FBDK, which was provided by Rockwell Automation. The authors would like to thank Dr. J.H.Christensen for his fruitful suggestions.

7. REFERENCES

1. Christensen, James H. "Basic Concepts of IEC 61499"; Distributed Automation 2000.
2. IEC 65/248/PAS, Function blocks for industrial-process measurement and control systems - Part 1: Architecture, 19 April 2000.
3. Sasajima, Hisashi. "Fieldbus, The Digital Wave in Measurement, Control and Networks", ICAM ASIA 2001.
4. Tsunematsu, Hirotsugu. "IEC61499 Distributed Function Block Concept", ICAM ASIA 2001.