Control and Data Flow Testing on Function Block Diagrams

Eunkyoung Jee, Junbeom Yoo, and Sungdeok Cha

Department of Electrical Engineering and Computer Science,
Korea Advanced Institute of Science and Technology(KAIST)
and AITrc/IIRTRC/SPIC,
373-1 Guseong-dong, Yuseong-gu, Daejeon, Republic of Korea
{ekjee, jbyoo, cha}@dependable.kaist.ac.kr

Abstract. As programmable logic controllers(PLCs) have been used in safety-critical applications, testing of PLC applications has become important. The previous PLC-based software testing technique generates intermediate code, such as C, from function block diagram(FBD) networks and uses the intermediate code for testing purposes. In this paper, we propose a direct testing technique on FBD without generating intermediate code. In order to test FBD, we define testing granularity in terms of function blocks and propose an algorithm that transforms an FBD network to a flow graph. We apply existing control and data flow testing coverage criteria to the flow graph in order to generate test cases. To demonstrate the effectiveness of the proposed method, we use a trip logic of BP(Bistable Processor) at RPS(Reactor Protection System) in DPPS(Digital Plant Protection System) which is currently being developed at KNICS[1] in Korea.

1 Introduction

Software testing is the act of exercising software with test cases for the purpose of finding failures [2]. Because failures of safety critical software can cause serious damage to life or property, testing of safety critical software has become an indispensable step required to assure software quality.

In the nuclear power plant control system, as existing analog systems have been replaced by digital systems controlled by software, testing of digital control systems has become more important. The control software is usually implemented on PLCs which are widely used to implement safety critical real-time systems. To test PLC applications, the characteristics of PLC programming languages should be considered. This work focuses on the FBD which is one of the most widely used standard PLC programming languages.

A PLC application implemented by FBD is automatically compiled to PLC machine code and executed on PLC. Testing of PLC machine code is difficult due to its complexity. Although the behavior of FBD is similar to the procedure or function of procedural program languages, there is no systematic way to apply software testing techniques to FBD. In previous cases[3], FBD testing

R. Winther, B.A. Gran, and G. Dahll (Eds.): SAFECOMP 2005, LNCS 3688, pp. 67–80, 2005.

has been done on intermediate C source code transformed from FBD networks. Although this method can test FBD networks at some level, it cannot be applied to FBD networks from which intermediate C code cannot be generated. Moreover, generating intermediate code leads to additional cost.

In this paper, we propose a direct cost-efficient testing method on FBD without generating intermediate code. We assume that the transformation process from FBD to PLC machine code has no errors. Because the transformation process has been validated for several decades by many PLC vendors, this assumption is reasonable. First, we define granularity of FBD testing. FBD is composed of network of function blocks. We define unit and module of FBD from the perspective of a function block network. In this paper, we focus on unit testing of FBD. To execute FBD unit testing, we propose an algorithm for the transformation of an FBD network to a flow graph. After generating a flow graph from an FBD network, we apply existing control and data flow testing strategies to the flow graph. To demonstrate the effectiveness of the proposed method, we use a trip logic of BP at DPPS RPS which is being currently developed at KNICS[1] in Korea.

The remainder of the paper is organized as follows: section 2 briefly introduces FBD and software testing, and section 3 defines granularity of FBD testing. In section 4, we propose an algorithm to transform an FBD network to a flow graph. We apply control and data flow testing strategies to the flow graph transformed from a real FBD example in section 5. Finally, conclusion and future works are described in section 6.

2 Background

2.1 Function Block Diagram

A PLC[4] is an industrial computer widely used in control systems such as chemical processing systems, nuclear power plants or traffic control systems. A PLC is an integrated system that consists of a CPU, memory, and input- and output-points.

IEC 61131-3[5] identifies PLC programming languages, which includes Structured Text(ST), Function Block Diagram(FBD), Ladder Diagram(LD), Instruction List(IL), and Sequential Function Chart(SFC). FBD is one of the most widely used PLC languages. FBD is easy to understand and good for representing data flow between control blocks.

FBD represents system behaviors by means of signal flow among function blocks. Functions between input variables and output variables are configured by a network of function blocks in the form of a circuit. Function blocks figured by rectangles are connected by input variables on the left side and output variables on the right side. Function blocks are classified into several groups according to their functions.

Figure 1 shows several function block groups and an example function block of each group. RPS, which currently being developed at KNICS[1], is programmed with function blocks which belong to the five function block groups in figure 1.