# 고장수목을 이용한 Function Block Diagram의 위험성 분석 기법 연구

이동아○, 김의섭, 유준범
건국대학교 컴퓨터공학부
{ldalove, atang34, jbyoo}@konkuk.ac.kr

# Hazard Analysis of Function Block Diagram using Fault Tree

Dong-Ah Lee○, Eui-Sub Kim, Junbeom Yoo
Konkuk University, Division of Computer Science and Engineering

**요 약**

안전필수 시스템 (Safety Critical System) 에서 소프트웨어가 차지하는 비중이 점점 커짐에 따라, 안전필수 시스템의 소프트웨어에 대한 위험성 분석 (Hazard Analysis) 이 크게 요구되고 있다. 원자력 분야와 같은 안전필수 시스템에서 제어를 위해 사용하는 디지털 제어 시스템 중, Programmable Logic Controller (PLC) 는 Function Block Diagram (FBD) 으로 설계한 소프트웨어를 사용한다. 이와 같이 안전필수 시스템에서 사용되는 소프트웨어 중 FBD를 이용해 설계한 소프트웨어의 위험성 분석을 위한 기법에 대한 연구는 많이 진행되지 않은 상황이다. 본 논문에서는 FBD를 이용해 설계한 소프트웨어의 위험성 분석을 위하여 고장수목 (Fault Tree, FT) 을 이용한 분석방법을 소개한다. 프로그램의 출력 중 위험으로 분류되는 값의 원인을 분석하기 위하여 고장수목을 이용해 원인을 찾아내는 기법이다. Function Block (FB) 의 유형별로 분석하는 방법이 다르기 때문에 각 유형별로 적용할 수 있는 고장수목 템플릿을 소개하며, FB 유형 중 시간과 관련된 연산을 수행하는 FB을 분석하기 위하여 Temporal Gate를 사용한 템플릿도 소개한다.

## 1. 서 론

위험성 분석은 안전한 시스템을 개발하기위하여 수행되는 여러 활동에게 위험 요소를 제거하거나 활동의 질을 높이기 위해 많은 정보를 제공해준다 [1]. 이러한 위험성 분석 기법 중 흔히 사용되는 기법인 고장수목 분석 (Fault Tree Analysis, FTA) 은 고장의 원인을 분석하기 위한 기법이다. 고장수목 (Fault Tree, FT) 은 Event와 Gate를 사용해서, 상위 Event (결과) 의 발생을 Gate를 이용한 하위 Event (원인) 의 조합으로 표현하는 하향식 (Top-down) 분석기법이다 [2]. FTA는 하드웨어의 위험성을 분석하기위해 처음 개발되었으며, 소프트웨어의 위험성을 분석하기 위한기법도 개발되었다 [3].

원자력이나 항공, 철도와 같은 안전필수 시스템(Safety Critical System)은 사고 시 중요한 경제적 손실이나 물질적 피해, 혹은 사람의 생명에 위협이 될 수 있다 [4]. 안전필수 시스템에서 소프트웨어가 차지하는 비중이 높아짐에 따라, 소프트웨어의 안전성이 전체 시스템의 품질을 결정하는 중요한 요소 중 하나가 되었다.

원자력 발전소 제어시스템 분야에서 사용하는 산업용 컴퓨터인 Programmable Logic Controller (PLC) 는 제어를 위한 소프트웨어를 개발하여 PLC 위에서 동작하는 구조를 가진다. 이 제어 소프트웨어는 IEC 61131-3 에서 제공하는 PLC 용 표준 프로그램 언어 5가지를 (LD, FBD, ST, IL, SFC) 사용해 개발한다 [5]. PLC에서 동작하는 소프트웨어에 대한 위험성 분석은 해당 언어에 대한 특성을 이해하고, 분석의 결과는 정확하면서도 효과적으로 알아볼 수 있도록 제시되어야 한다. 본 논문에서는 PLC에서 동작하는 소프트웨어 중 FBD를 사용해 개발한 소프트웨어를 FTA를 이용하여 위험성 분석을 하는 기법을 소개한다.

하드웨어에 대한 FTA는 고장이나 위험의 원인이 확률이나 장비의 노후도, 발생 빈도 등과 같은 요소들이 원인으로 분석되는 사례들이 많다. 따라서 분석가가 가지는 해당 분야에 대한 전문성과 분석 방법의 이해도에 따라서 결과가 달라진다. 반면 소프트웨어의 고장이나 위험은 예상치 못한 입력 값에 대한 결과로 나타난다. 따라서 분석 기법이 체계적으로 확립된다면 고장이나 위험으로 판단되는 출력에 대한 원인, 즉 입력의 조합을 정확히 찾아낼 수 있으며, 분석의 자동화도 가능하다.

FBD로 설계된 소프트웨어의 위험성 분석을 위해 본 논문에서는 FB 분류별 FT 템플릿을 제공한다. FB가 가질 수 있는 입력의 조합을 찾아낼 수 있도록 FT의 요소들을 활용해 템플릿을 개발하였다. 템플릿은 위험성 분석의 자동화가 가능하도록 고려하여 개발하였다. FB 분류 중 타이밍 계산이 가능한 FB의 분석을 위해 Temporal Gates [6] 를 포함한 템플릿을 개발하였다.

## 2. 고장수목 템플릿

본 장에서는 5가지 (Type Conversion, Numerical, Bit-String, Selection and Comparision, Timer) 로 분류한 FB 각각에 대한 FT 템플릿을 상세히 설명한다. 5가지 분류는 IEC 61131-3에서 정의한 표준 FB들 중 분석이 가능

한 FB만을 선별하여 정의하였다.

### 2.1. Type Conversion FBs

Type Conversion FBs은 입력으로 들어온 자료의 형식을 특정 형식으로 변환하기 위해 사용하는 FB로서 <그림 1>과 같이 나타낸다. 출력을 만들어 낼 수 있는 입력을 찾아내기 위하여 입력이 가질 수 있는 모든 경우의 수를 찾아내기 위한 FT 템플릿을 개발하였다.


그림 1 Type Conversion FB의 Graphical Form

<그림 2>는 Type Conversion FBs를 위한 FT 템플릿 중 *REAL_TO_INT* FB용 템플릿을 나타내고 있다. 정수형으로 변환된 출력 값을 낼 수 있는 입력 값을 알아내기 위한 템플릿이다. 입력 값은 특정 정수로 변환될 수 있는 실수 값의 범위로 나타난다.

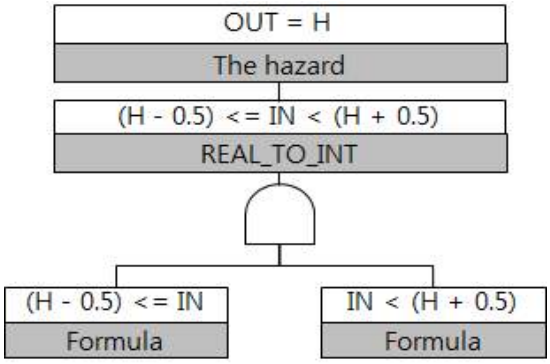


그림 2 *REAL_TO_INT* FB용 FT 템플릿

### 2.2 Numerical FBs

Numerical FBs은 산술연산 및 로그, 삼각함수 등의 연산을 위한 FB이다. Numerical FBs는 최소 한 개 이상의 입력을 가지는 FB로서, 로그나 삼각함수 등의 연산을 수행하는 FBs는 한 개의 입력과 한 개의 출력을 가지며, 산술연산을 수행하는 FBs는 최소 한 개 이상의 입력을 가지며 한 개의 출력을 가진다. 한 개의 입력을 가지는 수학연산 FB의 Graphical Form은 <그림 1>과 유사하며, Block 내부의 이름만 다른 형태를 가진다.

<그림 3>은 Numerical FBs 중 절댓값을 계산하는 *ABS* FB에 대한 FT 템플릿을 나타낸다. 출력 OUT의 값이 H가 되는 입력의 경우를 찾아내기 위한 템플릿으로, 입력에 대한 분석 결과는 두 가지 값으로 나타난다.

### 2.3 Bit-string FBs

Bit-string FBs는 Bit-shift연산이나 Boolean연산 (*AND*, *OR*, *XOR*, *NOT*) 을 위한 FB이다. *NOT* FB를 제외한 FBs는 하나 이상의 입력을 가질 수 있으며, <그림 4>와

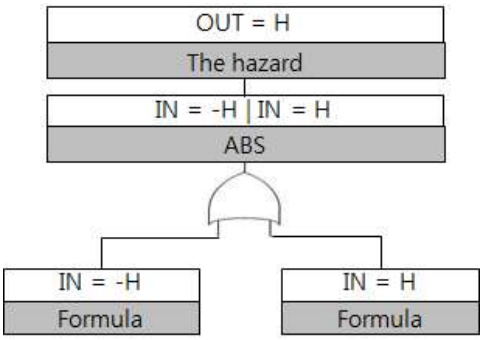


그림 3 *ABS* FB용 FT 템플릿

같은 Graphical Form을 가진다. 그림의 *Name*부분에 FB의 이름을 표기하여 연산의 종류를 나타낸다.

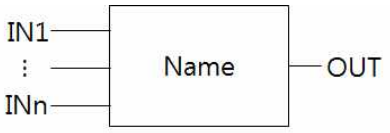


그림 4 n개의 입력을 가지는 FB의 Graphical Form

<그림 5>는 두 개의 입력을 가지는 *AND* FB 의 FT 템플릿을 나타내고 있다. 출력 OUT의 값이 True인 경우와 False인 경우에 대한 입력 값들을 분석하기 위한 형태이다. 입력의 수에 따라 분석 결과로 나오는 Event들의 개수가 정해진다.

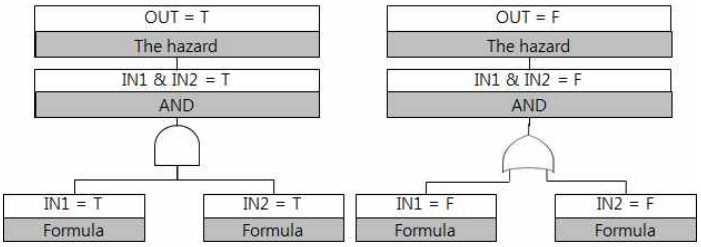


그림 5 입력이 두 개인 *AND* FB용 FT 템플릿

### 2.4 Selection and comparison Functions

입력으로 들어온 값을 선택적으로 사용하기 위한 FB 중 *SEL* FB가 <그림 6>에 나타나있다. 입력 *IN1*에 따라 출력 값이 *IN2* 또는 *IN3*으로 결정되는 연산을 수행하는 FB로서, 총 세 개의 입력과 한 개의 출력을 가진다.

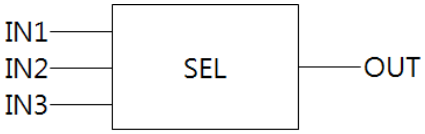


그림 6 *SEL* FB의 Graphical Form

*SEL* FB의 출력의 원인을 알아내기 위한 템플릿이 <그림 7>에 나타나있다. 출력 값이 IN2가 되는 경우와 IN3이

되는 모든 경우에 대하여 분석이 가능한 형태이며, 모든 입력이 가질 수 있는 값을 결정적으로 분석해 준다.

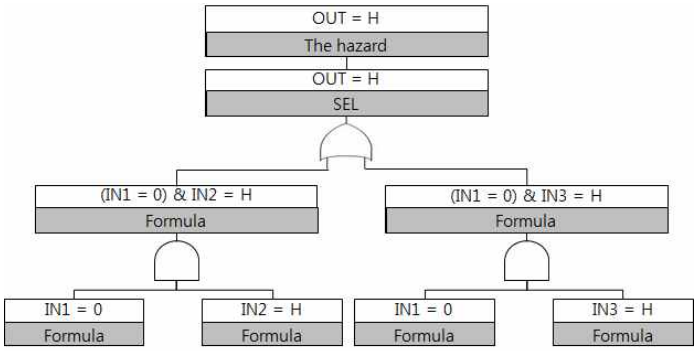


그림 7 *SEL* FB용 FT 템플릿

2.5 Timers FBs

Timers FB는 지금까지 소개된 FBs와는 성격이 다르다. 입력으로 들어온 신호가 특정 시간동안 유지되는지 확인하여 출력 값을 내보내주는 역할을 한다. <그림 8>은 Timer FBs중 *TON* FB의 Graphical Form을 나타내고 있으며, 해당 FB는 두 개의 입력과 두 개의 출력을 가진다. *IN2* 시간 동안 '1' 값을 *IN1*로 받게 되면 *OUT1*로 '1' 값이 출력되는 기능을 하는 FB로서, *IN2*에 의해 시간을 조건으로 가지는 역할이다.

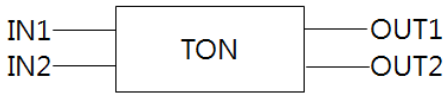


그림 8 *TON* FB의 Graphical Form

전통적인 FT는 시간을 다룰 수 있는 Event나 Gate가 존재하지 않는다. 따라서 [5]에서 제안한 Temporal Gates 중 일부를 이용해 Timer FBs를 분석 가능하도록 템플릿을 고안하였다. *FORPAST N* Gate는 N 시간동안 하위 이벤트가 만족해야하는 것을 의미하는 Gate이다.

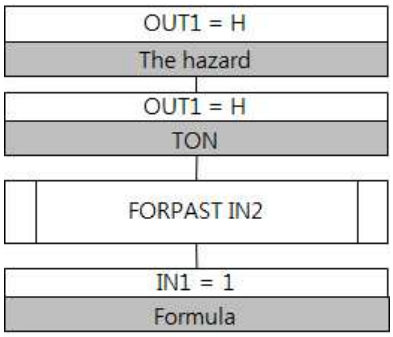


그림 9 TON FB용 FT 템플릿

## 3. 결론 및 향후 연구

안전필수시스템에서 사용할 수 있는 프로그램 중, FBD를 사용하여 설계한 프로그램의 위험성 분석하기 위한 기법을 제안하기 위하여, 고장수목 분석기법을 이용한 위험성 분석기법을 소개하였다. 고장수목 요소들을 사용하여 FB 분류 별 템플릿을 제공하였으며, 전통적인 고장수목 요소만을 이용해서는 특정 FB들에 대한 분석이 불가능하여 Temporal Gate를 이용한 템플릿을 추가로 개발하였다. 본 논문에서 제시한 FT 템플릿을 이용한 FBD로 설계된 소프트웨어의 위험성을 분석 결과를 이용하여, 위험을 유발하는 출력 값에 대한 입력 값들의 조합을 찾아낼 수 있다.

전통적인 고장수목 분석기법은 전문가에 의해 이루어지지만, 소프트웨어에 적용할 시 정확한 규칙과 정의가 제공된다면 분석의 자동화가 가능하다. 본 논문에서 제안한 템플릿들을 이용해 FBD로 설계한 프로그램에 대한 고장수목 분석을 자동으로 수행할 수 있는 도구를 구현할 것을 계획하고 있다.

## 4. 사사

본 연구는 2012 년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (2012-0003619).

## 참 고 문 헌

[1] Nancy Leveson, "SafeWare: system safety and computers," Addison-Wesley, 1995

[2] U.S.NRC, "Fault Tree Handbook (NUREG-0492)," 1981

[3] Nancy G. Leveson, Peter R. Harvey, "Software fault tree analysis," Journal of Systems and Software, Volume 3, Issue 2, Pages 173-181, 1983

[4] Ian Sommerville, "Software Engineering 8th," Addison-Wesley, 2008

[5] IEC (International standard for programmable controllers): Programming languages 61131-Part 3, 1993

[6] Girish Keshav Palshikar, "Temporal fault trees," Information and Software Technology, Volume 44, Issue 3, pp. 137-150, 2002