

흐름 제어 언어의 통합분석을 위한 확장 ST

*정은영, 김선주, 김태완, 장천현, 김문희
건국대학교 컴퓨터공학과
e-mail : eyjung@cse.konkuk.ac.kr

EST for Analysis of Flow Control Language

Eun-young Jeong, Sun-Ju Kim, Tae-Wan Kim, Chun-Hyon Chang, Moon-Hea Kim
Dept. of Computer Engineering, Kon-Kuk University

요 약

제어 시스템에 사용되는 흐름 제어 언어로는 IL(Instruction List), ST(Structured Text), FBD(Function Block Diagram), SFC(Sequential Function Chart), LD (Ladder Diagram)가 있다. 일반적으로 제어 시스템에 탑재하여 사용하는 언어는 상기 언어 중 두 종류 이하의 특정 언어로 제한되어 있다. 이러한 제약을 보완하기 위해, 모든 흐름 제어 언어를 통합 분석할 수 있는 통합 분석기가 필요하다. 본 논문에서는 흐름 제어 언어의 통합 분석 처리가 가능하도록 그래픽 표현의 FBD 를 문자 표현의 EST(Extended Structured Text)로 변형하는 규칙과 문자 표현의 EST 를 IL 로 변형하는 규칙을 제시한다. 언어간의 변형 과정에서 FBD 를 ST 로 표현할 수 없는 부분을 EST 로 정의한다. 또한, 본 논문에서 제안된 EST 를 기반으로 통합 분석기의 구조를 제시한다.

1. 서론

오늘날 자동화 시스템은 다양한 산업현장에서 널리 사용되고 있다. 자동화 시스템을 구성하기 위한 제어 장치는 PLC(Programmable Logic Controller), RTU(Remote Terminal Unit)이며, 시스템은 DCS(Distributed Control System), SCADA(Supervisory Control and Data Acquisition) 등이 있다. 제어 시스템에 사용되는 프로그래밍 언어로는 IL, ST, FBD, SFC, LD가 있다. 이러한 흐름 제어 언어는 산업용 표준인 IEC(International Electrotechnical Commission) 61131-3 에서 제안한 것이다[1].

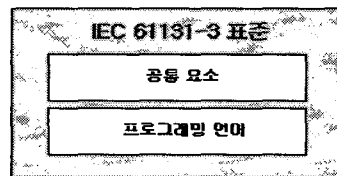
본 논문에서는 흐름 제어 언어에 대한 특성을 설명한 후 이를 토대로 그래픽 표현의 FBD 를 문자 표현의 EST 로 변형하는 방법과 문자 표현의 EST 를 IL 로 변형하는 방법을 제시한다. 또한, FBD 를 ST 로 표현할 수 없는 부분은 EST 에 정의함으로써 언어간의 변형을 가능하게 하였다. 마지막으로, EST 를 기반으로 한 통합 분석기의 구조를 제시한다.

본 논문의 구성은 다음과 같다. 2 장에서는 관련연구로 IEC 6131-3 에서 제안한 표준 언어 IL, ST, FBD, SFC, LD 에 대해 설명한다. 3 장에서는 FBD 를 EST 로 변환하는 규칙과 EST 를 IL 로 변환하는 규칙을 제시한다. 4 장에서는 기존의 분석기가 갖는 문제점을 살펴보고 EST 를 적용한 통합 분석기의 구조를 제시한다.

2. 관련연구

2.1 IEC 61131-3 표준

흐름 제어 언어에 대한 규약은 IEC 61131-3 에 나타나 있다. 각 언어의 공통화된 표현 형태를 공통 요소에 정의하고 각각의 흐름 제어 언어에 대한 사양은 프로그래밍 언어부분에서 정의하고 있다[2].



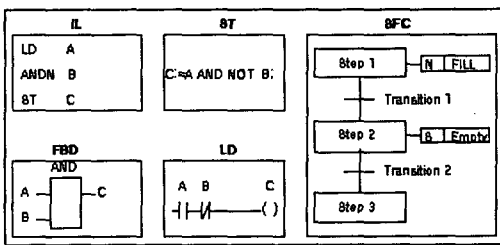
(그림 1) IEC 61131-3 구성

공통 요소에는 각 언어에서 범용으로 사용할 수 있는 데이터 타입과 변수에 대하여 정의되어 있다. 데이터 타입은 논리형, 정수형, 실수형, 문자열형, 사용자가 정의한 데이터 타입 등 일반 데이터를 표현하기 위해 다양한 형태로 정의하고 있다. 변수는 변수를 선언한 부분에서만 사용 가능하며 같은 이름으로 다른 부분에서 재사용 가능한 지역 변수와 전체에서 참조할 수 있는 전역 변수로 구성되어 있다[3].

FBD	LD	SFC	IL	ST
그래픽형		문자형		
표준 프로그래밍 언어				

(그림 2) IEC 61131-3 표준 언어

프로그래밍 언어부분에서는 종류별 언어에 대한 구성 및 개념에 대하여 설명하고 있다. IEC 61131-3에서는 언어에 대한 공통화 사양에 대해 정의한 표준 프로그래밍 언어 부분과 문자 기반의 언어인 IL 과 ST 그리고 그래픽 기반의 LD 와 FBD 를 구분하여 설명한다[3].



(그림 3) IL, ST, FBD, LD, SFC 의 예

2.2 흐름 제어 언어

흐름 제어 언어는 (그림 2)에서와 같이 그래픽형과 문자형으로 구분된다. 그래픽형 언어는 (그림 3)에서 처럼 FBD, LD 로 구분되며 문자형 언어는 IL, ST 로 구분되어진다. SFC 는 타 언어에서 정의한 로직의 흐름을 정의하는데 사용된다.

다음은 각 흐름 제어 언어의 특징이다.

IL 은 컴퓨터 언어의 어셈블리 언어와 유사한 형태를 취하며, 연산자, 변경자, 피연산자 등의 단순화된 언어 형태를 취하고 있다[4].

ST 는 컴퓨터 언어인 Pascal 또는 Basic 과 매우 유사한 형태의 고급 언어이다. 컴퓨터 언어에서 기본 구조인 조건, 순환 처리에 대한 내용을 모두 포함하고 있다[5].

FBD 는 산업 처리용에서 흔히 사용되는 것으로 기능과 기능의 동작으로 표현되며, 제어 요소들 간의 정보나 데이터의 흐름을 나타내기에 적합하다. 주기적인 작업 수행을 정의하기 위해 사용되는 언어이다. 또한 전자회로처럼 그래픽 블록형태의 조합을 취하고 있다[4].

LD 는 코일이나 접점 등의 릴레이 로직을 그래픽 형태로 표현하며, 모션, 연결선, 접점, 코일 등의 구성 요소를 보유한다[6].

SFC 는 타 언어에서 표현된 작업들에 대한 상위 구조로서의 개념을 가지며, 전체 수행에 대한 흐름 정의에 적합하다. 구성요소로는 스텝, 천이, 액션, 액션 제한자, 흐름선 등이 있다[4].

3. 언어 변환 규칙

이번 장에서는 통합 분석기를 통해 처리되는 흐름 제어 언어간의 변환 규칙을 제시한다. 흐름 제어 언어 중 그래픽 표현의 FBD 를 문자 표현의 EST 로 변환시키는 규칙과 EST 를 IL 로 변환시키는 규칙을 제시한다.

(표 1) FBD-EST 연산자 변환 규칙

구분	FBD	EST
1		B:=A;
2		C:=(A < B);
3		C:=(A1 & A2 & ...An-1 & An);

3.1 FBD 와 EST 변환 규칙

FBD 와 EST 에서 공통으로 사용되는 연산자, 함수, 함수 블록간의 변환 규칙을 각각의 특성에 따라 구분하여 제시하였다.

(표 1)은 FBD 가 갖는 입력인자와 출력인자의 수에 따라 변환 규칙을 제시하였다[7]. 입력 인자와 출력 인자가 각각 하나인 경우, 2 개의 입력인자와 1 개의 출력인자를 갖는 경우, N 개의 입력인자와 1 개의 출력인자를 갖는 경우로 구분하여 변환 규칙을 적용하였다. N 개의 입력인자를 갖는 경우는 피연산자의 수가 2 개 이상인 경우로 피연산자의 수가 증가함에 따라 입력인자의 수도 증가된다.

(표 2)와 (표 3)은 함수와 함수 블록에 대한 변환 규칙이다[7]. (표 1)의 연산자 변환 규칙에서처럼 함수와 함수 블록을 구성하는 입력인자와 출력인자의 수에 따라 변환 규칙을 제시하였다.

(표 2) FBD-EST 함수 변환 규칙

구분	FBD	EST
1		B:=COS(A);
2		C:=MAX(A,B);
3		D:=INSERT(A,B,C);
4		E:=REPLACE(A,B,C,D);

(표 2) FBD-EST 함수 변환 규칙

구분	FBD	EST
5		F:=MUX4(A,B,C,D,E);
6		J:= MUX8(A,B,C,D,E,F,G,H,I);

(표 3) FBD-EST 함수 블록 변환 규칙

구분	FBD	EST
1		R_TRIG(A); B:=R_TRIG.Q;
2		RS(A,B); C:=RS.Q1;
3		CMP(A,B); LT:=CMP.LT; EQ:=CMP.EQ; GT:=CMP.GT;
4		AVERAGE(A,B,C); XOUT:=AVERAGE.XOUT;
5		CTD(A,B,C); D:=CTD.Q; E:=CTD.CV;
6		INTEGRAL(A,B,C,D,E); F:=INTEGRAL.Q; G:=INTEGRAL.XOUT;
7		CTUD(A,B,C,D,E); F:=CTUD.Q1; G:=CTUD.Q2; H:=CTUD.CV;

3.2 EST와 IL 변환 규칙

이번 장에서는 3.1 장에서 변환 규칙을 적용하여 변환된 EST를 IL로 변환하는 규칙을 문장 형식에 따라 구분하여 설명한다.

(표 4)는 연산 및 형 변환에 대한 변환 규칙이다[7].

(표 4) EST-IL 연산자 변환 규칙

ST	IL	ST	IL
a1 := a2	LD a2, ST a1	ANY_TO_BOOL(a1)	LD a1, ANY_TO_BOOL
a1 + a2	LD a1, ADD a2	ANY_TO_DINT(a1)	LD a1, ANY_TO_DINT
a1 - a2	LD a1, SUB a2	ANY_TO_REAL(a1)	LD a1, ANY_TO_REAL
a1 / a2	LD a1, DIV a2	ANY_TO_SINT(a1)	LD a1, ANY_TO_SINT
a1 < a2	LD a1, LT a2	ANY_TO_STRING(a1)	LD a1, ANY_TO_STRING
a1 <= a2	LD a1, LE a2	ANY_TO_TIME(a1)	LD a1, ANY_TO_TIME
a1 <> a2	LD a1, NE a2		
a1 = a2	LD a1, EQ a2		
a1 >= a2	LD a1, GE a2		
a1 > a2	LD a1, GT a2		
-(a1)	LD a1, MUL -1		

(표 5)와 (표 6)은 EST의 제어문과 조건문을 IL로

변경하기 위한 변환 규칙이다.

(표 5) EST-IL 제어문 변환 규칙

종류	EST	IL
FOR문	FOR a1:=a2 TO a3 BY a4 DO Statements_1 END_FOR	LD a2 ST a1 StarLevel: LD a1 LE a3 ANY_TO_BOOL JMPN EndLevel LD a1 ADD a4 ST a1 Statements_1 JMP StarLevel EndLevel:
WHILE문	WHILE LogicalExp_1 DO Statements_1 END_WHILE;	StarLevel: LogicalExp_1 ANY_TO_BOOL JMPN EndLevel Statements_1 JMP StarLevel EndLevel:

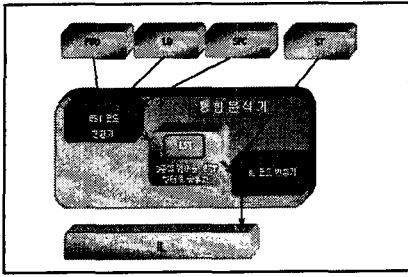
(표 6) EST-IL 조건문 변환 규칙

종류	EST	IL
IF문	IF LogicalExp_1 THEN Statements_1 ELSE Statements_2 ENDIF	LogicalExp_1 ANY_TO_BOOL JMPN Level_1 Statements_1 Level_1: Statements_2
CASE문	CASE a1 OF Num_1:Statements_1 Num_2:Statements_2 Num_3:Statements_3 ELSE Statements_4 END_CASE	LD a1 EQ Num_1 JMP Level_1 LD a1 EQ Num_2 JMP Level_2 LD a1 EQ Num_3 JMP Level_3 JMP Level_4 Level_1: Statements_1 JMP EndLevel Level_2: Statements_2 JMP EndLevel Level_3: Statements_3 JMP EndLevel Level_4: Statements_4 EndLevel:

4. 통합 분석기

본 장에서는 IL, ST, FBD, SFC, LD를 동시에 처리할 수 있는 통합 분석기의 구조를 제시한다. 통합 분석을 위해서는 고급언어에 속하는 ST, FBD, SFC, LD를 어셈블리 언어와 유사한 IL로 변환시키는 과정이 필요하다.

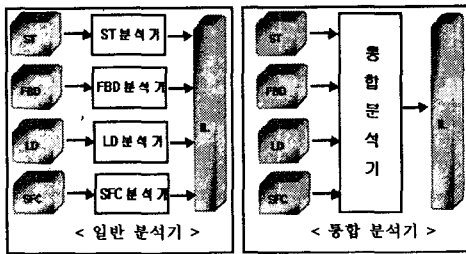
변환 과정은 다음과 같다. FBD, SFC, LD를 EST로 변환 시킨 후에 이를 다시 IL로 변환 시킨다. 이때 중간언어로 일반 프로그램언어와 표현이 유사한 EST를 이용한다. FBD, SFC, LD를 ST로 표현할 수 없는 부분은 ST를 확장한 EST를 정의하여 IL로 변환이 가능하도록 하였다.



(그림 4) 통합 분석기 구조도

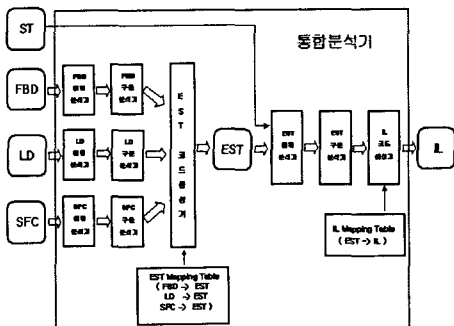
4.1 통합 분석기 구조

기존에 사용되고있는 분석기의 구조와 통합 분석기의 구조를 비교하여 살펴 보겠다. 기존의 분석기는 단일 구조로 되어있다. 즉, 특정 언어 하나에 대한 분석만을 담당한다. 이와 반대로, 통합 분석기는 (그림 5)에서 보듯이 다중적인 구조로서 ST, FBD, LD, SFC 언어를 동시에 분석할 수 있다.



(그림 5) 일반 분석기와 통합 분석기 구조 비교

통합 분석기의 내부 구조는 (그림 6)과 같다. FBD, LD, SFC는 EST 코드 생성기를 거쳐 EST로 변환되며 변환된 EST는 IL 코드 생성기를 거쳐 IL로 변환된다. EST 코드 생성을 위해 FBD, LD, SFC와 EST 간의 변환 규칙에 따른 매핑 테이블을 작성하여 이용한다. 또한, IL 코드 생성기는 EST와 IL 간의 변환 규칙에 따른 매핑 테이블을 이용하여 IL로의 변환이 가능하도록 하였다.



(그림 6) 통합 분석기의 내부 구조

4.2 통합 분석기의 필요성 및 장점

기존의 언어 분석기는 ST, FBD, LD, SFC, IL 중 일부 언어만을 선택적으로 분석 처리할 수 있도록 설계되어 있다. 그러나, 이를 보완한 통합 분석기를 이용함으로써 단일 소프트웨어 기반에서 ST, FBD, LD, SFC, IL의 분석 처리가 가능하게 되었다.

통합 분석기는 IL, ST, FBD, SFC, LD 언어를 동시에 처리할 수 있으므로 소프트웨어 모듈을 단일화, 단순화 시키므로 소프트웨어 크기를 감소시킨다. 또한, 시뮬레이션 후 디버깅을 용이하게 한다.

통합 분석기는 매핑 테이블을 이용하여 그래픽 형태의 고급언어(FBD, LD, SFC)를 텍스트 형태의 고급언어(EST)로 이를 다시 저급언어(IL)로 변환시켜 준다. 따라서, 고급언어(FBD, LD, SFC)를 고급언어(EST)로 변환 시켜주는 과정을 분리하여 설계한다. 하지만, EST를 IL로 변환하는 과정은 동일하므로 새로운 설계를 필요로 하지 않는다.

5. 결론

본 논문에서는 흐름 제어 언어 중 가장 많이 사용하는 그래픽 표현의 언어 FBD를 문자 표현의 언어 EST로 변환하는 규칙과 EST를 IL로 변환하는 규칙을 제시하였다. 제시한 EST를 기반으로 통합 분석기의 구조를 제안하였으며 이를 통해 IL, ST, FBD, LD, SFC의 통합 분석이 가능하도록 하였다. 또한, EST를 기반으로 단일 소프트웨어를 사용함에 따라 통합 시뮬레이션 개발이 용이하게 되었으며, 기존 시스템에서 할 수 없었던 흐름 제어 언어의 결합 시뮬레이션이 가능하게 되었다. 통합 분석기는 제어 감시 시스템에 기본 탑재하여 상하수처리장, 전력감시 시스템, 모니터링 시스템 등의 산업분야에서 사용될 수 있다.

참고문헌

- [1] <http://www.autotech.co.kr/new/0105/special/0501.htm>
- [2] International Standard IEC 61131-3, First Edition, 1993
- [3] <http://www.plcopen.org>
- [4] 리얼게인, RealPLC1.0 manual. 2001
- [5] <http://www.software.rockwell.com/corporate/reference/iec1131/st.cfm>
- [6] <http://www.performancesw.com/pawhelp>
- [7] ISaGRAF PRO User Guide, 1999