

흐름 제어용 그래픽 언어 통합 처리

정구희, 김태완, 장천현
건국대학교 컴퓨터정보통신공학과
e-mail:koohi@cse.konkuk.ac.kr

Unified Processing of Graphical Languages for Sequence Control

Koo Hi Chung, Tae Wan Kim, Chun Hyon Chang
Dept of Computer Science and Engineering,
KonKuk University

요 약

산업 분야에서 자동화 시스템은 제품의 자동설계, 생산 공정의 자동제어, 장애의 발견과 복구, 품질 검사 등 각종 생산과 관련되어 생산성을 향상시킨다. 자동화 시스템을 구성하는 제어 시스템에 사용되는 흐름 제어 언어에는 IL, ST, FBD, SFC, LD의 5종이 있다. 일반적인 소프트웨어에서 동시에 사용할 수 있는 흐름 제어 언어의 종류는 2종 이하로 제한되어 있다. 이에 본 논문에서는 그래픽 기반의 흐름 제어 언어인 FBD, LD, SFC의 3종을 통합하여 처리가 가능한 소프트웨어를 제시한다. 그래픽 기반의 흐름 제어 언어인 FBD, LD, SFC의 형식을 모두 포함하는 데이터 구조를 설계하고, 데이터 구조로부터 읽어 들인 데이터를 출력하는 그래픽 처리 방안을 제시한다. 소프트웨어에서 편집한 FBD, LD, SFC는 파일 단위로 관리할 수 있고, 각 파일들은 프로젝트 파일로 통합하여 관리가 가능하다.

1. 서론

오늘날 다양한 산업 현장에서 제품의 자동설계, 생산 공정의 자동제어, 생산설비의 관리, 장애의 발견과 복구, 품질검사 등에 자동화 시스템을 사용함으로써 생산성을 향상시키고 있다. 이러한 자동화 시스템을 구성하기 위한 제어 시스템에 사용되는 언어로는 IEC61131-3에서 제안한 IL(Instruction List), ST(Structured Text), FBD(Function Block Diagram), SFC(Sequential Function Chart), LD(Ladder Diagram)의 5종이 있다. 이 중 IL과 ST는 문자 기반의 언어이고, LD와 FBD는 그래픽 기반의 언어이며, SFC는 문자와 그래픽 기반의 언어이다. 일반적으로 사용하는 소프트웨어에서는 사용할 수 있는 흐름 제어 언어의 종류가 2종 이하로 제한되어 있다.

본 논문에서는 그래픽 기반의 흐름 제어 언어인 FBD, LD, SFC의 3종을 모두 포함하는 데이터 구조를 설계하고, 데이터 구조로부터 읽어 들인 정보를 그래픽을 사용하여 처리함으로써 3종의 흐름 제어

언어를 통합하여 처리가 가능한 소프트웨어의 구현을 제시한다.

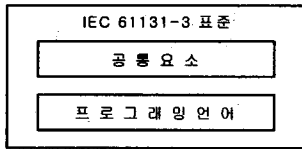
본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 IEC61131-3에서 제안한 흐름 제어 언어에 대한 규약과 표준 언어인 IL, ST, FBD, SFC, LD에 대한 구성 및 개념을 설명한다. 3장에서는 소프트웨어의 전체 설계 구조를 제시한다. 그래픽 기반의 흐름 제어 언어인 FBD, LD, SFC를 모두 포함하는 데이터 구조를 구성하고, 데이터 구조로부터 읽어 들인 정보를 이용한 그래픽 처리 방법을 제시한다. 구현된 소프트웨어는 흐름 제어 언어인 FBD, LD, SFC의 3종을 통합하여 처리할 수 있고, 관리가 가능하다.

2. 관련 연구

2.1 IEC 61131-3 표준 규약

IEC 61131-3에서는 흐름 제어 언어에 대한 규약을 제안하고 있다. (그림 1)과 같이 전체적인 구성은 공통 요소와 프로그래밍 언어의 두 부분으로 되어

있다.



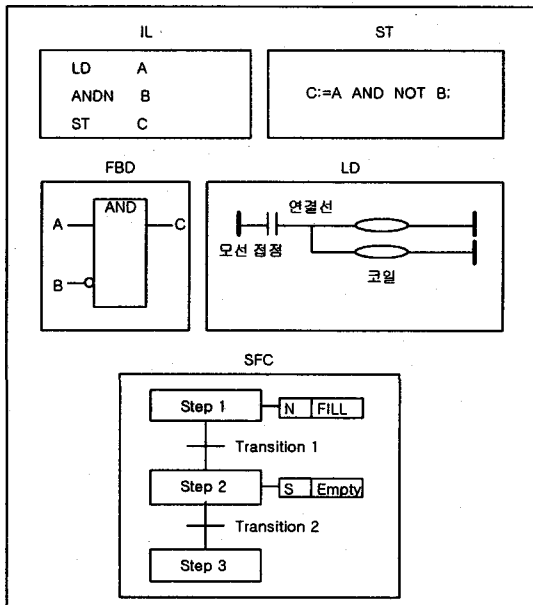
(그림 1) IEC 61131-3 구성

공통 요소에는 각 언어에서 범용으로 사용하는 데이터 타입과 변수를 정의하고 있다. 데이터 타입은 논리형, 정수형, 실수형, 문자열형, 사용자 정의 데이터 타입 등 일반 데이터의 표현을 위해 다양한 형태로 정의되어 있다[1]. 변수는 선언한 부분에서만 사용 가능하며 같은 이름으로 다른 부분에서 재사용 가능한 지역변수와 전체에서 참조 가능한 전역변수로 정의되어 있다.

프로그래밍 언어에서는 종류별 흐름 제어 언어의 구성 및 개념에 대하여 정의하고 있다.

2.2 흐름 제어 언어

흐름 제어 언어는 그래픽 기반의 언어인 LD와 FBD, 텍스트 기반의 언어인 IL과 ST, 그래픽과 텍스트 기반의 언어인 SFC로 구성되어 있다[2].



(그림 2) IL, ST, FBD, LD, SFC의 예

IL은 컴퓨터 언어인 어셈블리 언어와 유사한 형태로써 연속된 명령어로 이루어져 있으며, 명령어 리스트로 구성되어 있다. 연산자, 변경자, 피연산자 등의 단순화된 언어 형태를 취하고 있다[4].

ST는 컴퓨터 언어인 파스칼 또는 베이직과 유사한 형태를 가지고 있다. 자동화 과정에 활용되는 고급 언어이며, 컴퓨터 언어의 기본 구조인 조건, 순환, 처리에 대한 내용을 모두 포함하고 있다. 데이터 조작과 수학적 계산을 수행할 수 있는 구조와 명령어 집합으로 되어 있기 때문에 주요 장점은 데이터 핸들링에 있다[4].

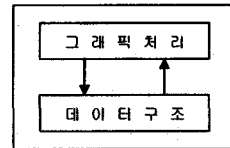
FBD는 산업 처리용에서 흔히 사용되고 있다. 기능과 기능의 동작으로 표현되며, 제어 요소들 간의 정보나 데이터의 흐름을 나타내기에 적합하다. 주기적인 작업 수행을 정의하는 언어로 사용되며, 전자 회로처럼 그래픽 블록 형태의 조합을 취하고 있다[4].

LD는 코일이나 접점 등의 릴레이 로직을 그래픽 형태로 표현하여 데이터를 검사하고 변경하게 한다. 구성 요소로는 모션, 연결선, 접점, 코일 등이 있다[4].

SFC는 다른 언어에서 표현된 작업들에 대한 상위 구조로서의 개념을 가지며, 전체적인 수행에 대한 흐름 정의에 적합하다. 래더 프로그램의 대체언어로 대두되고 있으며 이산 제어 시스템의 순차 논리를 그래픽 적으로 표현할 수 있다. 제어 프로그램의 순차적 논리를 명확하게 표시하기 때문에 프로그램의 유지, 작성, 보수와 판독이 용이하다는 장점을 갖고 있다. 스텝, 천이, 액션, 액션 제한자, 흐름선 등으로 구성되어 있다[4].

3. 설계 방안

3.1 전체 설계 구조



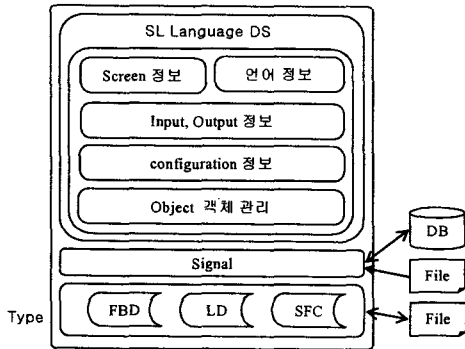
(그림 3) 전체 구조

전체 구조는 (그림 3)과 같이 그래픽 기반의 흐름 제어 언어인 LD, FBD, SFC의 데이터 구조 모듈과 그래픽 처리 모듈로 구성되어 있다. 데이터 구조 모듈로부터 필요한 정보가 들어 있는 객체의 포인터를 언어 그래픽 처리 모듈에서 정보를 처리하여 입출력을 하게 된다. 데이터 구조와 그래픽 처리 기능의 모듈화를 통하여 처리할 수 있는 흐름 제어 언어의 확장을 가능하게 하였다. 소프트웨어의 개발상 독립을 가능하게 하여 시뮬레이션을 할 수 있게 하였고,

데이터 구조를 분리하여 산업용 임베디드 시스템에 재사용이 가능하게 하였다. 그래픽 처리가 데이터 구조의 영향을 받지 않게 함으로써 그래픽의 기능 개선을 가능하게 하였다.

3.2 데이터 구조

데이터 구조는 (그림 4)와 같이 각 흐름 제어 언어의 형식에 관한 데이터, 시그널 데이터, 흐름 제어 언어의 정의에 관한 데이터로 구성되어 있다. 흐름 제어 언어의 형식에 관한 데이터에 FBD, LD, SFC를 모두 포함하여 단일 데이터 구조에서 흐름 제어 언어 3종을 구성할 수 있게 하였다.



(그림 4) 데이터 구조

흐름 제어 언어의 형식에 관한 정보는 파일에서 읽어 들임으로써 파일의 수정을 통해 언어의 기능을 조정할 수 있게 하였다. 읽어 들인 흐름 제어 언어의 형식에 관한 정보는 신호 정보와 자동으로 결합되어 수행이 가능하게 하였다.

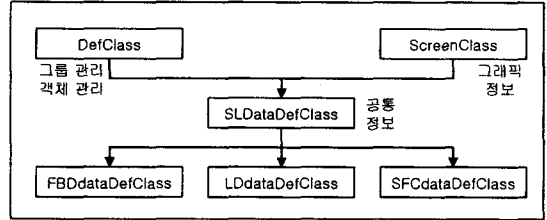
신호 정보의 수집은 단일 감시 제어 시스템에서 사용할 때에는 파일 입출력을 통해 가능하게 하였고, 대규모 감시 제어 시스템에서 사용할 때에는 신호 정보가 포함된 데이터베이스 서버로부터 입출력을 할 수 있게 하였다.

흐름 제어 언어의 형식에 관한 정보와 신호 정보는 흐름 제어 언어의 구성 요소에 관한 정보와 결합하여 그래픽 처리 모듈에서 처리하게 하였다.

데이터 구조의 처리는 (그림 5)와 같이 계층 구조로 되어 있어 단위 객체 구조의 상속만으로 FBD, LD, SFC의 데이터 구조를 처리 가능하게 하고 있다.

DefClass는 그래픽 기반 언어인 FBD, LD, SFC의 그룹 관리와 객체 관리를 하며, ScreenClass는 FBD, LD, SFC의 구성 요소가 출력되는 화면에 관한 정보를 가지고 있다. SLDataDefClass는 흐름 제

어 언어의 공통 정보를 처리하고 있다. SLDataDefClass의 하위 객체들은 각 FBD, LD, SFC의 구성 요소에 관한 정보를 가지고 있다.

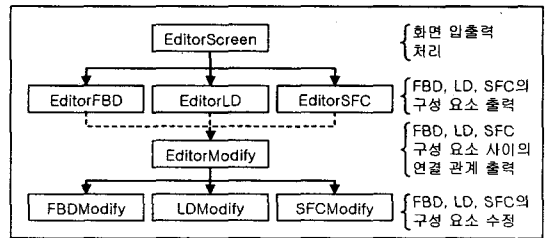


(그림 5) 데이터 구조 처리

DefClass에서의 그룹관리는 흐름 제어 언어의 종류와 무관한 처리가 가능하게 하였고, 객체 언어별 형식과 무관한 데이터 연결을 할 수 있게 하였다. 데이터 연결은 Memory 허용 가능치 까지 가능하게 하였고, FBD, LD, SFC의 혼용을 할 수 있게 하였다.

3.3 그래픽 처리

데이터 구조에 저장된 정보를 이용하여 FBD, LD, SFC의 구성 요소를 입출력하는 모듈은 (그림 6)과 같이 계층 구조로 되어 있어 처리되는 흐름 제어 언어의 확장을 가능하게 하였다.

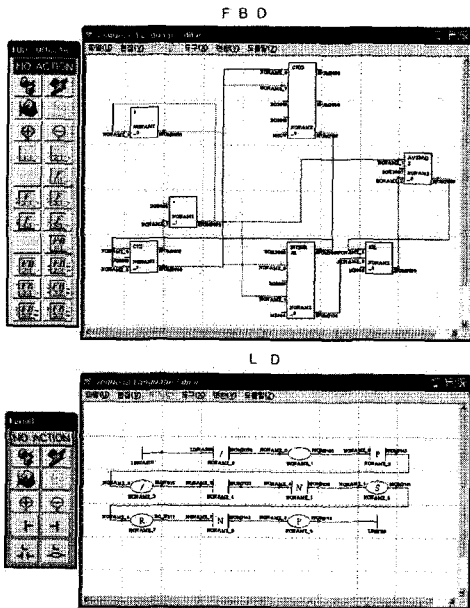


(그림 6) 그래픽 처리

EditorScreen에서 FBD, LD, SFC의 구성 요소에 대한 화면 입출력을 처리하게 하여 하위 객체에서 처리된 정보가 출력되는 대상을 바꿀 수 있게 하였다. EditorFBD, EditorLD, EditorSFC는 출력되는 각 FBD, LD, SFC 구성 요소의 그래픽 처리에 관한 정보를 가지고 있고, 추가된 구성 요소에 관한 정보를 데이터 구조 모듈에 보내게 하였다. EditorModify객체에는 하위 객체에서 사용하는 공통 기능이 정의되어 있고, 상위 객체들의 포인터를 가져오게 하였다. 구성 요소 사이의 연결 관계 출력을 EditorModify객체에서 하게 함으로써 각 FBD, LD,

SFC 구성 요소를 수정하여 발생하는 연결 관계의 변화를 출력할 수 있게 하였다. EditorModify객체의 하위 객체들은 각 FBD, LD, SFC 구성 요소의 이동, 삭제, 복사에 관한 처리를 하고, 구성 요소의 수정에 관한 정보를 데이터 구조 모듈에 보내게 하였다.

(그림 7)과 같이 FBD, LD의 구성 요소를 입출력 할 때에는 격자 단위로 할 수 있게 하여 구성 요소 단위의 편집이 가능하게 하였다. 편집하는 화면의 격자수를 조정함으로써 입출력되는 구성 요소의 수를 조정할 수 있게 하였고, 화면의 확대와 축소를 통해 하나의 화면에서 전체 구성 요소를 입출력할 수 있게 하였다. 편집하는 흐름 제어 언어의 종류에 따라 서로 다른 입력 인터페이스를 사용하여 각 흐름 제어 언어의 구성요소를 입출력할 수 있게 하였다. 구성 요소를 수정하는 인터페이스의 일관성을 유지하여 구성 요소를 동일한 방법으로 수정할 수 있게 하였으며, 구성 요소 사이의 연결 관계를 설정하는 인터페이스의 일관성을 유지하여 구성 요소 사이의 관계를 동일한 방법으로 설정하게 하였다.

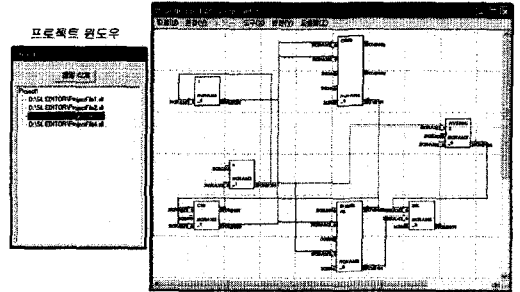


(그림 7) FBD와 LD의 입출력
FBD, LD, SFC는 개별 파일 단위로 입출력이 가능하게 하였고, 프로젝트 파일에서 FBD, LD, SFC 파일을 통합하여 관리할 수 있게 하였다.

(그림 8)과 같이 프로젝트 윈도우를 사용함으로써 FBD, LD, SFC 파일의 관리를 가능하게 하였으며,

파일 단위의 입출력에 사용한 인터페이스를 사용하여 인터페이스의 일관성을 유지하였다.

FBD, LD, SFC 파일의 입출력 정보와 프로젝트 파일의 정보는 데이터 모듈로 보내지게 하였다.



(그림 8) 프로젝트 윈도우의 사용

4. 결론

본 논문에서는 흐름 제어 언어인 FBD, LD, SFC, ST, IL의 5종 중 그래픽 기반의 언어인 FBD, LD, SFC의 3종을 통합하여 처리할 수 있는 데이터 구조와 그래픽 처리 방안을 설계 및 구현하였다. 제시한 데이터 구조와 그래픽 처리를 기반으로 그래픽 기반의 흐름 제어 언어인 FBD, LD, SFC의 3종을 단일 소프트웨어 상에서 설계하여 구현할 수 있게 하였다. 단일 소프트웨어의 사용으로 통합 시뮬레이션의 개발이 가능하게 되었으며, 기존 시스템에서 불가능했던 흐름 제어 언어의 결합 시뮬레이션이 가능하게 되었다. 구현된 소프트웨어는 제어 감시 시스템에 기본 탑재하여 상하수 처리장, 전력감시 시스템, 모니터링 시스템 등의 산업 분야에서 사용될 수 있다.

참고문헌

- [1]International Standard IEC61131-3, First Edition, 1993
- [2]PLCopen 홈페이지, <http://www.plcopen.org>
- [3]ISaGRAF PRO User Guide 1999
- [4]리얼게인 홈페이지, <http://222.realgain.co.kr>
- [5]정은영, 김선주, 김태완, 장천현, 김문희, 흐름 제어 언어의 통합분석을 위한 확장 ST, 한국정보처리학회 학술발표논문집 9권 제1호 2002, pp. 1013-1016
- [6]김선주, 김태완, 장천현, 흐름 제어 언어 분석 도구 설계 및 구현, 한국정보과학회 학술발표논문집 29권 제2호 2002, pp. 634-636