

Virtual Factory를 이용한 제어 프로그램의 가시적 검증

아주대학교 | 박상철

1. 서론

현대의 제조 기업은 끊임없이 변화하는 시장 환경에서 비교 우위의 생존 경쟁력을 유지하기 위해 새로운 제품의 개발뿐 만 아니라 지속적인 생산 기술의 향상을 추구하여야 한다[1]. 따라서 효율적인 생산 시스템의 신속한 구축은 제조 기업의 경쟁력을 좌우하는 매우 중요한 이슈이다. 최근의 생산 시스템은 용접 및 물류 이송 로봇, 컨베이어 벨트, ASRS(Automated Storage & Retrieval System) 등의 자동화된 설비와 이를 제어하는 컴퓨터로 구성된 고 차원의 통합 시스템이다. 이러한 특성을 가지는 생산 라인의 구축에는 막대한 자본이 투자되기 때문에 시스템 초기 설계 과정에서의 결정은 매우 신중하게 행해져야 한다.

시뮬레이션은 수리적으로 기술되기 어려운 복잡한 시스템의 설계와 분석에 유용한 도구이다[2]. 시뮬레이션 기술은 설비의 활용도(Utility)를 통계적으로 산출하거나 병목지점 찾기, 제조 스케줄 작성 및 오류 감지 등에 활용된다. 이러한 시뮬레이션 기술은 현재까지 학계와 산업 분야에 어느 정도 수용되었으나 생산 시스템의 세부 설계와 구축의 목적으로 사용될 만큼 실제적이지 못하기 때문에 생산 시스템의 설계 단계에서의 분석 도구로 남아 있다. 하나의 예로 실제 생산 시스템은 일반적으로 Programmable Logic Controller(PLC) 프로그램에 의해 제어 및 운용되지만 기존의 시뮬레이션은 각 프로세스 사이의 독립적인 개체의 흐름으로 제어 로직을 대략적으로 묘사한다.

생산 라인의 세부 설계를 위해서는 시스템의 통계적인 생산성 측면뿐만 아니라 상호 운용되는 설비의 물리적 유효성(Validity)과 효율성(efficiency), 그리고 제어 프로그램의 타당성을 포함하는 실제적인 시뮬레이션이 요구된다. 그림 1에서 보이는 PLC는 산업 공정 제어 기술로써 가장 적합하며 광범위하게 채택되고 있다. 즉각적으로 반응하는 평행 회로의 작동을 묘사하기 위해 PLC는 입, 출력 신호와 순환적 스캔 싸

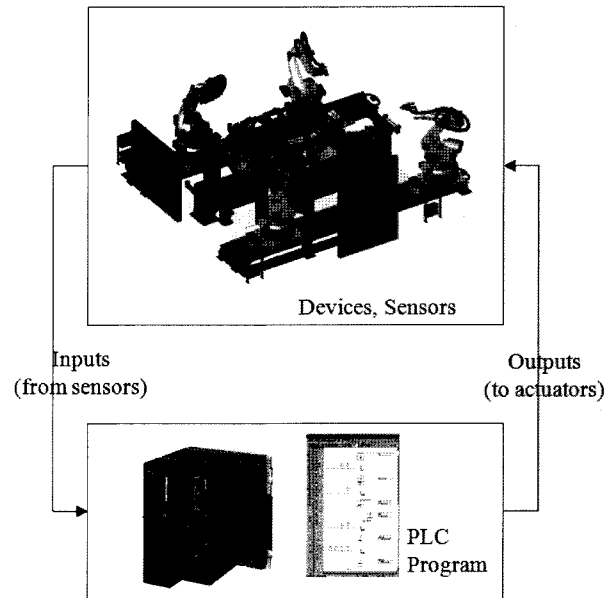


그림 1 Production system controlled by a PLC program

이를 사용한다. 공정 상황에 대한 감지 신호인 PLC 입력 신호는 프로그램 스캔 사이클마다 갱신되며 갱신된 공정 정보에 따라 PLC 내부의 Boolean 로직을 해석하여 이에 따른 PLC출력 신호를 통하여 다음 공정을 명령하게 된다. PLC에서 이 Boolean 논리는 일반적으로 Ladder Diagram 이라는 그래픽 언어로 표현된다[3].

PLC 시스템의 모델링 및 검증을 위한 기존의 소프트웨어는 UPPAAL2k, KRONOS, Supremica 그리고 Hy-Tech 같은 Timed Automata 방법론에 기반하여 개발되었으며 이는 주로 Statement List 언어의 PLC 프로그램 검증에 사용된다[4]. 그러나 기존 PLC 프로그램 검증 소프트웨어는 주로 이론적인 속성(안전, 시스템 생존)에 초점을 맞춘 제한적인 검증이며 사용자가 원하는 실제 시스템의 제어 목적을 달성하는가의 여부를 확인하기가 어렵다.

본 글의 목적은 가시적인 PLC 프로그램 검증을 가능하게 하는 PLC 검증 아키텍처를 소개하는 것이다.

소개하는 PLC 검증 환경에서는 사용자가 좀 더 쉽고 직관적으로 PLC 프로그램을 검증할 수 있도록 가상 설비로 구성된 가상 공장모형을 채택한다. 본 글의 전체적인 구성은 다음과 같다. 2장에서는 소개하는 PLC 프로그램 검증 아키텍처에 대하여 기술하며 3장은 PLC 프로그램에 동시에 대응하는 공장 모델의 효율적인 구축 방법론에 대해 설명한다. 4장에서는 예제 공정에 대한 적용과, 5장에서는 결론이 기술된다.

2. PLC 프로그램의 가시적 검증

PLC 프로그램의 효율적 검증 아키텍처를 설계하기 위해서는 PLC 프로그램이 기존에 어떤 과정으로 작성되는가를 이해하는 것이 중요하다. Chuang et al.[5]은 자동화 생산 시스템의 구축에 대한 9단계의 절차에 대해 다음과 같이 기술하였다.

- Step 1) 제어 대상 공정 정의
- Step 2) 공정 운전 스케치
- Step 3) 세부 공정 순서 정의
- Step 4) 각 공정의 순차적 수행을 위해 필요한 센서의 추가
- Step 5) 프로세스 셋업 혹은 공정 체크를 위해 필요한 매뉴얼 컨트롤 삽입
- Step 6) 작업 근로자의 안전 사항을 고려 후 필요 시 추가 및 조정 장치 만들기
- Step 7) 안전 비상 정지를 위한 마스터 스톱 스위치 추가
- Step 8) PLC 프로그램의 기초로 사용 될 Ladder 로직 다이어그램 만들기
- Step 9) 공정 순서가 정상에서 벗어날 가능한 지점의 고려.

위 과정 중, 공정 제어 기술자에게 가장 시간 소모적인 과정은 Step 8이며, 이 단계에서는 보통 의도하는 제어 목적이 달성 될 때 까지, 코드 작성, 수정, 디버깅의 반복적인 방법에 의해 수행된다[6]. PLC 프로그램 작성 과정에서 공정 제어 기술자는 대상 시스템이 운전 중, 각 공정이 임의 특정 상태에 놓일 것이라는 가정과 상상으로 PLC 코드를 작성하게 되기 때문에 코드 작성에 많은 시간이 소요되며 또한 작성자의 실수에 의한 오류를 포함시키게 될 확률이 높게 된다. 더욱이 설비의 설치가 완료된 후 부분적인 공정을 운영하는 방법으로 진행되는 PLC 코드 검증 과정에서 실제 설비의 충돌과 같은 사고의 위험이 산재하며 자동화 양산 체제에 이르는 전체 시간의 지연이 발생한다. 이러한 이유로 인하여 기존의 PLC

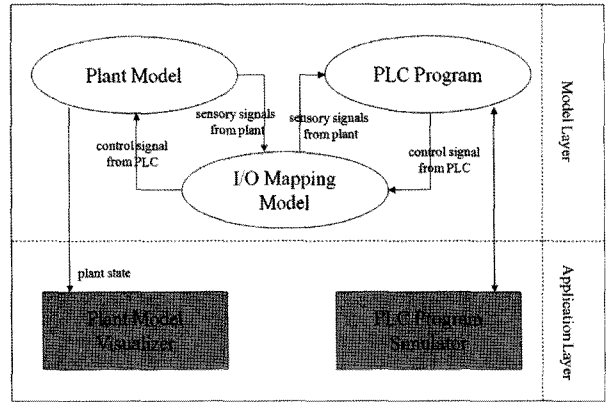


그림 2 The proposed PLC Programming environment

프로그램 환경은 많은 시간이 소요되는 비 효율적인 과정이다. 생산 라인의 규모와 커지고 복잡도가 증가함에 따라 좀더 효율적인 PLC 프로그램 검증 환경이 절실히 요구되는 원인이 여기에 있다.

그림 2는 본 글이 소개하는 PLC 프로그램 검증 아키텍처를 보여준다. 모델 층(model layer)과 응용 층(application layer)의 두 계층으로 구성된 아키텍처에서 모델 층은 공장모형(virtual factory model), PLC 프로그램(control model) 그리고 I/O 매핑(I/O mapping)의 세 가지 모델로 구성된다. 공장 모델은 PLC 출력 신호에 따라 작동하는 설비의 구동 프로그램뿐만 아니라 설비의 형상과 운동학(Kinematic) 정보를 포함한 모든 면에 대한 정보를 포함하며, PLC 프로그램을 이런 공장모형을 제어하는 로직을 나타낸다. I/O 매핑 모델은 공장 모델과 PLC 프로그램이 신호를 주고받으며 서로 통신하기 위한 모델이다. 응용계층에서는 사용자를 위한 두 가지 인터페이스를 제공한다. 'PLC Simulator'는 제어 프로그램의 시뮬레이션을 수행하며 'Plant Model Visualizer'는 PLC 시뮬레이션 동안 생산 시스템의 상태 변화를 반영하여 대응하는 공장 모델(3D 그래픽 모델)의 움직임을 보여준다. 따라서 사용자는 Plant Model Visualizer를 통하여 직관적인 방법으로 PLC 프로그램 검증이 가능하다.

모델 계층의 세 가지 모델 중, 공장 모델은 본 글이 소개하는 PLC 프로그램 검증 환경에서 중요한 역할을 담당한다. 앞서 언급했듯 공장 모델은 PLC 출력 신호에 따른 설비의 구동 프로그램뿐만 아니라 형상과 운동학 정보를 비롯한 설비의 모든 면에 대한 정보를 포함한다. 따라서 공장 모델은 실제와 같이 컴퓨터상의 가상공간에서 제조 공정을 수행하는 'virtual factory model'로 여겨진다. 가상 공장의 구현을 위해 실제 공장의 모든 물리적인, 논리적인 요소에 대하여 디지털화된 모델을 구축하는 것이 필요하다.

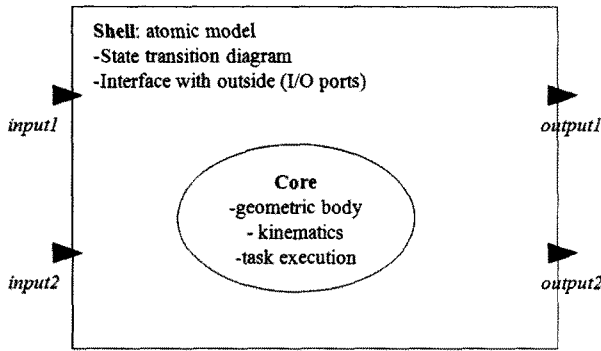


그림 3 Virtual device model

공장모델에서는 레이아웃 상에 각 위치를 가지는 제조 설비들로 구성된다. 제조 설비를 표현하기 위하여 실제 설비의 물리적, 논리적인 관점을 반영하는 디지털 모델인 가상 설비 모델 개념을 채택한다. 가상 설비 모델은 설비의 형상과 운동학 정보를 포함하는 설비의 본질적 속성뿐만 아니라 PLC의 신호에 특정 공정을 수행한다는 논리적 관점 또한 포함하여야 한다. 이를 위하여 가상 설비 모델을 외형(Shell)와 핵심(Core)의 두 부분으로 분리한다. 외형 부분은 설비 모델의 중복적 사용을 위하여 다양한 생산 시스템에 유동적으로 적용 가능하도록 하며 핵심 부분은 설비가 가진 본질적 관점의 정보를 포함시킨다. 그림 3은 가상 설비 모델의 개념을 보여 준다.

가상 설비는 생산 시스템의 다양한 구성에 적용이 가능하여야 함으로 가상 설비의 재 사용성은 매우 중요하다. 만약 가상 설비의 재사용성이 만족되지 않는다면 PLC 검증을 위하여 특정 라인에 존재하는 동일한 설비에 대하여 반복적인 모델링 과정으로 설비 모델을 생성하여야 할 것이다. 이러한 설비의 재 사용성 관점을 만족시키기 위해 설비의 외형 파트는 다양한 형태의 생산 시스템에 사용가능 하도록 유연하여야 한다. 외형 파트의 유연함을 반영하기 위해 본 글에서는 Zeigler가 제안한 DEVS(Discrete Event System Specification) 형식론을 채택한다[7,8]. 이산 사건 모델의 계층적 정의와 모듈화된 도시를 위해 고안된 DEVS 형식론은 객체 지향 관점에서도 높은 양립성을 나타낸다.

DEVS 형식론에서는 대상 시스템을 두 가지 모델 형태로 나타낸다. 첫째는 원자모델(Atomic model)으로써 규모가 큰 상위 레벨의 모델을 구성하는 요소가 되는 기본 모델을 나타내며, 두 번째 조합모델(Coupled model)은 원자모델로 구성된 큰 단위의 모델을 나타내기 위해 조합모델 내에 포함된 원자모델이 서로 어떤 관계로 구성되었는가에 대하여 구조적으로 명시한다.

DEVS 형식론의 원자 모델 M 은 다음 7가지 세부 요소에 의해 명시된다.

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$$

X : input events set;

S : sequential states set;

Y : output events set;

$\delta_{int} : S \rightarrow S$: internal transition function;

$\delta_{ext} : Q^* X \rightarrow S$: external transition function

$Q = \{(s,e) \mid s \in S, 0 \leq e \leq t_a(s)\}$: total state of M ;

$\lambda : S \rightarrow Y$: output function;

$t_a : S \rightarrow Real$: time advance function

7가지 요소 중 네 가지 즉, δ_{int} , δ_{ext} , λ 그리고 t_a 는 원자 모델의 주요 함수이다. 두 번째 모델 형태인 조합 모델은 새로운 모델을 형성하기 위하여 모델을 구성하는 요소 모델의 연결 관계를 나타낸다. 조합 모델의 형식은 다음과 같이 정의 된다.

$$D_N = \langle X, Y, M, EIC, EOC, IC, SELECT \rangle$$

X : input events set;

Y : output events set;

M : set of all component models in DEVS;

$EIC \subseteq DN, IN^* M, IN$: external input coupling relation;

$EOC \subseteq M, OUT^* DN, OUT$: external output coupling relation;

$IC \subseteq M, OUT^* M, IN$: internal coupling relation;

$SELECT : 2^M - \Phi \rightarrow M$: tie-breaking selector,

여기서 확장자 $.IN$ 과 $.OUT$ 은 DEVS 모델의 입력, 출력 포트를 나타낸다. 공장 모델의 구축을 위해 가상 설비의 외형 파트는 원자 모델로 표현되며 전체 공장 모델은 원자 모델과 그들 사이의 관계를 정의하는 조합 모델로 표현될 수 있다. 공장 모델에 대한 자세한 설명이 다음 장에 이어진다.

3. 공장 모델(Plant Model) 구축

소개하는 PLC 프로그래밍 환경의 목적은 공장 모델과 PLC 프로그램을 연결함으로써 PLC 프로그램 검증을 위한 직관적인 환경을 제시하는 것이다. 위 목적의 달성을 위해서는 공장 모델의 효율적인 구축 절차를 도출하는 것이 필수적이다. 그림 4는 PLC 프로그래밍 환경에서 세 가지 모델 사이의 상호관계를 보여준다.

세 가지 모델은 공장 모델(Plant model), I/O 매핑

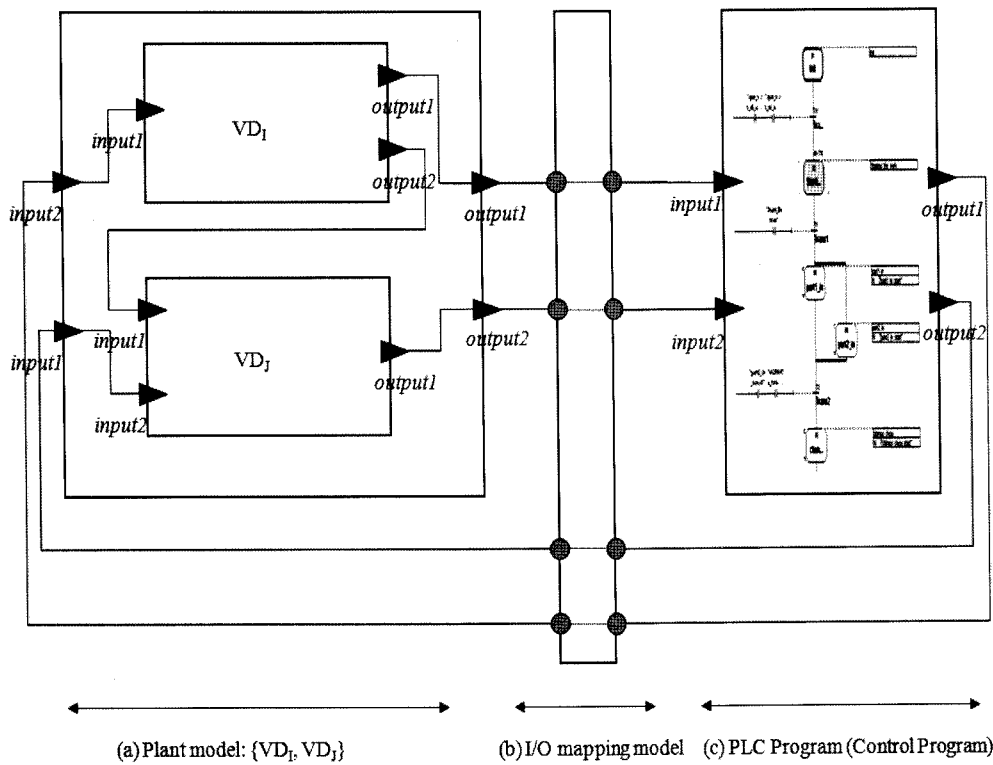


그림 4 Interaction among three models of the PLC programming environment

모델(I/O mapping model), 그리고 PLC 프로그래머 공장 모델은 I/O 매핑모델을 통하여 PLC 프로그램에 의해 제어된다.

공장 모델이 가상 설비들로 구성되었음을 고려하면, 공장 모델의 구축방법에 대한 설명에 앞서 가상 설비의 구축에 대한 설명이 선행되어야 할 것이다. 앞서 언급했듯 설비 모델은 외형과 핵심의 두 부분으로 구성된다. 설비의 형상과 운동학 그리고 설비 레벨의 명령 실행에 관한 정보를 포함하는 설비의 본질적 영역인 핵심 파트와 이 핵심 파트를 둘러싸며 가상 설비 모델이 다양한 생산 시스템의 구성에 유동적으로 적용되는 것을 가능하게 하는 외형 파트는 가상 설비 모델이 다양한 생산 시스템의 구성에 유동적으로 적용되는 것을 가능하게 한다. 이 부분이 Timed-FSA (Finite State Automata)에 기반을 두는 DEVS 형식론의 원자모델로 모델링 된다. 설비 모델의 외형파트를 모델링하기 위해서는 각 설비가 수행하는 일련의 작업을 정의하여야 하며 설비 작업은 일반적으로 PLC 출력이나 다른 설비 모델로부터 오는 외부 신호에 의해 야기된다. 일단 설비 모델의 작업셋이 정의되면 DEVS 형식론의 원자모델을 정의하는 상태 전이 다이어그램의 도출이 가능하다. 그림 5(a)는 T1(P1에서 P2 이동)과 T2(P2에서 P1이동)의 두 가지 작업을 가진 간단한 AGV(Automatic Guided Vehicle) 모델 예제를 보

여준다. 두 가지의 작업이 존재하므로 AGV의 외형 파트는 두개의 입력 포트를 가지며 그림 5(b)에 보이는 것처럼 *Signal_1*과 *Signal_2*로 표기된다. 작업 셋으로부터 상태 전이 다이어그램을 자동으로 생성하는 것이 가능하다. 본 AGV 예제는 *P1*, *DoT1*, *P2*, *DoT2*의 네 가지 상태를 가진다. *P1*과 *P2*는 상태 전이를 위해 입력 포트(*Signal_1*, *Signal_2*)로부터 오는 외부 신호에 의해 야기되는 반면, *DoT1*과 *DoT2*상태는 두 작업의 완료 이벤트(*T1*, *T2*)인 내부 이벤트에 의해 야기된다. AGV 가상 설비의 DEVS 원자 모델은 다음과 같다.

Shell of a virtual device: $M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$

- $X = \langle Signal_1, Signal_2 \rangle$
- $S = \langle P1, DoT1, P2, DoT2 \rangle$
- $Y = \langle T1Done, T2Done \rangle$
- $\delta_{int}(DoT1) = P2$
- $\delta_{int}(DoT2) = P1$
- $\delta_{ext}(P1, Signal_1) = DoT1$
- $\delta_{ext}(P2, Signal_2) = DoT2$
- $\lambda(DoT1) = T1Done$
- $\lambda(DoT2) = T2Done$
- $t_a(DoT1) = Time_1$
- $t_a(DoT2) = Time_2$

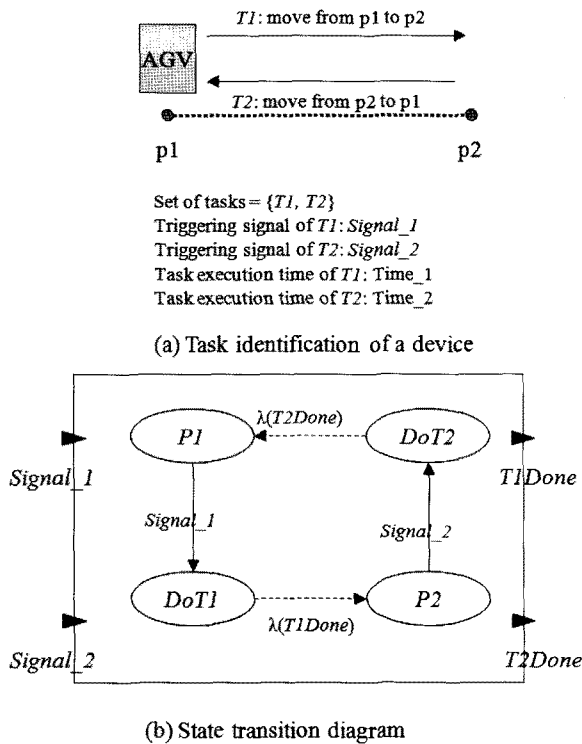


그림 5 Shell modeling of a virtual device

설비 모델이 구축되면 공장 모델은 가상 모델의 조합에 의해 정의된다. 가상 설비가 원자모델에 의해 표현되는 것과 달리, 공장 모델은 조합모델로 표현되며 이는 원자 모델과 이들 사이의 결합관계를 나타낸다. 그림 4(a)는 VD_i, DV_j의 두 가상 설비를 포함하는 단순한 공장모델 예제를 나타낸다. 공장모델의 DEVS 조합모델은 그림 4(a)와 같으며 다음과 같이 명시된다.

Plant Model :

$$DN = \langle X, Y, M, EIC, EOC, IC, SELECT \rangle$$

$$X = \{input1, input2\}$$

$$Y = \{output1, output2\}$$

$$M : = \{VD_i, VD_j\}$$

$$EIC = \{(DN, input1 * VD_i, input2), (DN, input2 * VD_i, input1)\}$$

$$EOC = \{(VD_i, output1 * DN, output1), (VD_j, output1 * DN, output2)\}$$

$$IC = \{(VD_i, output2 * VD_j, input1)\}$$

$$SELECT : 2^M - \Phi \rightarrow M: \text{tie-breaking selector.}$$

본 글에서 소개하는 공장 모델 구축 방법론의 이점은 크게 두 가지가 있다. 첫째는 가상 설비 모델의 재사용성으로 가상 설비 모델의 구조가 생산 시스템의 구성에 독립적으로 존재하는 것을 가능하게 한다. 두 번째 이점은 가상 설비 모델의 상태 전이 다이어그램

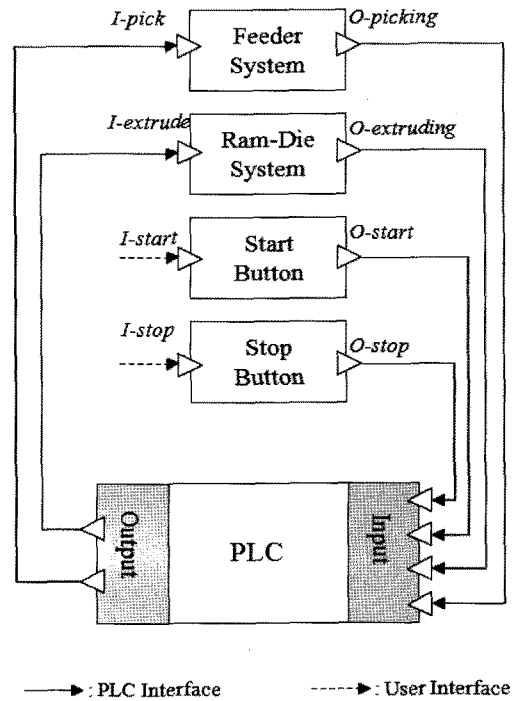


그림 6 Configuration of an extrusion system

이 직관적인 방법에 의해 정의되는 것이다. 이산 사건 모델에 대한 깊은 지식이 없는 사용자라 하더라도 설비가 취할 수 있는 작업 셋을 정의함으로써 간단하게 가상 설비 모델의 정의가 가능하다.

4. 예제 및 구현

제안하는 PLC 프로그래밍 환경의 프로토타입은 몇 개의 예제를 사용하여 테스트 및 구현 되었다. 그래픽 rendering을 위한 OpenGL과 함께 Visual Studio 환경의 C++ 언어가 사용되었다.

그림 6은 네 개의 하위 시스템으로 구성된 압출기 시스템 예제를 나타낸다. 각 하위 시스템은 1) Feeder system, 2) ram-die system, 3) start button, 4) stop button 로 구성된다. 압출 성형은 원하는 단면 형상을 얻기 위해 부드러운 상태의 재료 금속이 die의 입구로부터 밀어져 나오는 압축 공정이다. 압축 공정은 치약 입구로부터 치약이 밀려져 나오는 과정에 빗대어 묘사될 수 있다. 압출기 시스템의 제어 로직은 아래와 같다.

- 1) 시스템이 유ힴ(Idle) 상태일 때, 사용자가 Start Button을 누를 경우, PLC는 'I-pick'=TRUE의 값을 feeder system에 입력 값으로 전송한다.
- 2) 'I-pick'값이 TRUE가 될 때, feeder system은 재료가 되는 billet을 준비하는 작업을 시작한다. feeder system의 출력 포트 'O-picking'값은 이

- 과정이 끝날 때까지 TRUE 상태가 되며 준비 과정이 마무리 되면 'O-picking'은 FALSE가 된다.
- 만약 billet을 준비하는 과정이 종료되면('O-picking' = FALSE), PLC는 'I-pick' = FALSE, 'I-extrude' = TRUE의 두 값을 전송한다.
 - 'I-extrude'가 TRUE가 될 때, 압출 시스템은 die 입구로 금속재료(billet)를 밀어 넣기 시작한다. 압출 과정이 끝날 때 까지 압출기 시스템의 출력 포토 'O-extruding'은 TRUE가 된다. 압출 과정이 끝나면 'O-extruding'은 FALSE가 된다.
 - billet의 압출 과정이 끝나게 되면('O-extruding' = FALSE), PLC는 시스템 상태를 유휴(Idle)상태로 만들기 위해 'I-extrude' = FALSE를 전송한다.
 - 'I-stop' 신호가 사용자에게 의해 야기될 때마다, 시스템의 모든 상태는 정지 되어야 한다.

예제 시스템의 구현을 위해 압축기의 제어 로직은 PLC 프로그램으로 변환되며 그림 7은 그림 6 압출기 시스템을 제어하는 SFC(Sequential Function Chart) 언어로 표현된 PLC 프로그램을 나타낸다.

수직 라인을 따라 나열된 박스(S1, S2, S3, S4, S5)는 압출 시스템의 상태를 나타내며 상태 사이의 두꺼

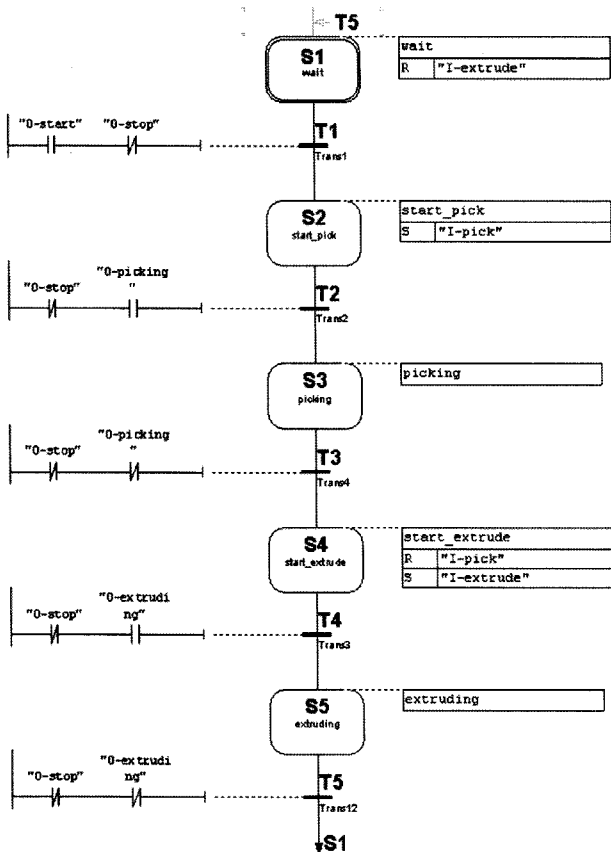


그림 7 PLC program that controls the extrusion system

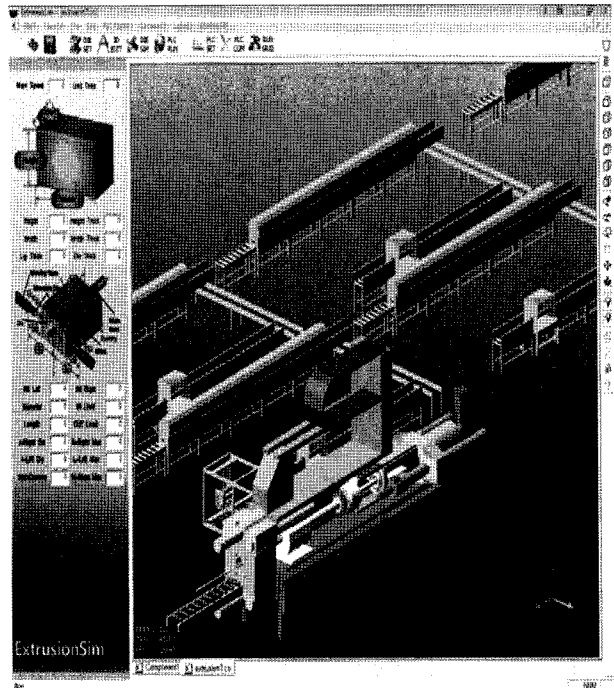


그림 8 PLC Simulation (PLC Studio®)

운 가로 줄 요소(T1, T2, T3, T4, T5)는 상태 전이를 의미한다. 각 상태 전이의 왼쪽에는 Ladder diagram이 보이는데 이는 상태 전이의 조건을 명시한다. 각 상태 박스의 오른쪽에 점선으로 연결된 박스는 압출 시스템을 제어하는 PLC에 의한 실행 명령(S: set, R: reset)을 나타낸다. 본 시스템의 PLC 프로그램이 가시적인 검증을 가능하게 하기 위해서는 공장 모델이 필요하다.

그림 8은 상업용 시스템인 PLC Studio®를 이용하여 만들어진 가상공장 모델이며, 실제 PLC 코드로 제어되고 있는 모습을 보여주고 있다.

5. 결론 및 토의

본 글에서는 PLC 프로그램과 이와 동시에 반응하는 공장 모델을 포함하여 가시적인 PLC 프로그램 검증을 가능하게 하는 PLC 프로그램 아키텍처에 대해 소개한다. 소개된 아키텍처의 모델 계층에는 공장 모델(virtual factory model), PLC 프로그램(control model), 그리고 I/O mapping의 세 가지 모델로 구성된다. 공장 모델은 생산 시스템의 모든 제조 설비를 포함하며 PLC 프로그램은 공장 모델에 대한 제어 로직을 내포한다. I/O 매핑 모델은 이 두 모델 사이를 연결하는 통신 역할을 한다. PLC 프로그래밍 환경에서 공장 모델은 매우 중요한 역할을 하기 때문에 가상 공장 모델 뿐만 아니라 가상 설비 모델의 구축에 실제적인 방법론의 도출이 필요하다. 이를 위해 DEVS

(Discrete Event System Specification) 형식론에 기반을 둔 공장 모델의 효율적인 구축 방법론을 소개하였다.

참고문헌

[1] B.K. Choi, B.H. Kim, New trends in CIM: Virtual manufacturing systems for next generation manufacturing, Current Advances in Mechanical Design and Production Seventh Cairo University Int. MDP Conf., Cairo, February 15-17, 2000, 425-436

[2] Klingstam P, Gullander P. Overview of simulation tools for computer-aided production engineering, Computers in Industry, 1999;38:173-186.

[3] Rullan A. Programmable logic controllers versus personal computers for process control, Computers and Industrial

[4] Manesis S, Akantziotis K. Automated synthesis of ladder automation circuits based on state-diagrams, 2005;36:225-233.

[5] Chuang C P, Lan X, Chen J C. A systematic procedure for designing state combination circuits in PLCs, Journal of Industrial Technology, 1999;15(3): 2-5.

[6] Manesis S, Akantziotis K. Automated synthesis of ladder automation circuits based on state-diagrams, 2005;36:225-233.

[7] B. P. Zeigler, Multifaceted modeling and discrete event simulation, Academic Press, Orland, 1984.

[8] T. G. Kim, DEVSIM++ User's Manual, Department of Electrical Engineering, KAIST, Korea, 1994.



박상철

1994 KAIST 산업공학과 학사
 1996 KAIST 산업공학과 석사
 2000 KAIST 산업공학과 박사
 2001~2004 미국 DaimlerChrysler ITM research engineer
 2004~현재 아주대학교 산업정보시스템공학부

교수

관심분야 : Digital Manufacturing System, Discrete Event System Modeling & Simulation, CAD/CAM
 E-mail : scpark@ajou.ac.kr

2009년 유비쿼터스 컴퓨팅 기술 및 응용 워크샵

- 일 자 : 2009년 7월 2일
- 장 소 : 제주그랜드호텔
- 주 최 : 한국정보과학회
- 문 의 : 조직위원장 홍봉희 교수 051-510-2424