

## 객체지향 설계 및 시뮬레이션을 이용한 자동 물류 핸들링 시스템의 제어 로직 검증

### Validation of the Control Logic for Automated Material Handling System Using an Object-Oriented Design and Simulation Method

한 관 희\*  
(Kwan Hee Han)

**Abstract** : Recently, many enterprises are installing AMSs(Automated Manufacturing Systems) for their competitive advantages. As the level of automation increases, proper design and validation of control logic is a imperative task for the successful operation of AMSs. However, current discrete event simulation methods mainly focus on the performance evaluation. As a result, they lack the modeling capabilities for the detail logic of automated manufacturing system controller. Proposed in this paper is a method of validation of the controller logic for automated material handling system using an object-oriented design and simulation. Using this method, FA engineers can validate the controller logic easily in earlier stage of system design, so they can reduce the time for correcting the logic errors and enhance the productivity of control program development. Generated simulation model can also be used as a communication tool among FA engineers who have different experiences and disciplines.

**Keywords** : PLC (Programmable Logic Controller), object-oriented design, object-oriented simulation, validation, AMS (Automated Manufacturing System), AHMS (Automated Material Handling System)

#### I. 서론

기업 환경의 급속한 변화로 인해 기업간 경쟁이 심화되면서 기업 생존의 전략적 도구로서 자동화 제조 시스템(automated manufacturing system)을 채택하는 기업이 점차 증가하고 있는 추세이며, 자동화 제조 시스템은 그 수행 기능에 따라 FMS (Flexible Manufacturing System), FAS (Flexible Assembly System), AS/RS (Automated Storage & Retrieval System), AMHS (Automated Material Handling System) 등으로 불리고 있다. 기업의 자동화 수준이 높아짐에 따라 공정 제어 방법과 물류 흐름은 더욱 복잡해지고 있으며 이에 따라 전체 시스템을 제어하는 제어기(controller)의 로직 설계 및 검증이 중요한 과제로 부각되고 있다.

현재 대부분의 자동화 제조 시스템 제어기로는 PLC (Programmable Logic Controller)가 채택되고 있으며 제어 로직은 래더 다이어그램(ladder diagram)이나 IL (Instruction List)등을 이용하여 작성된다. 제어기 로직을 구현하기 위한 현행 PLC 프로그래밍 과정은 체계적인 프로그램 설계 및 개발 프로세스 방법론의 적용이 미흡한 상태이며, 래더 다이어그램으로 대표되는 PLC 프로그래밍 언어는 프로그램 크기가 커지면 복잡도가 증가하여 이해하기 어렵고 이로 인해 사전 검증 작업에 많은 시간을 필요로 하게 된다.

그리고 최근 들어 생산 현장의 자동화 수준이 고도화되

고 복잡성이 증가하면서 새로운 자동화 제조 시스템의 설계 및 운영 효율성을 사전 점검하기 위한 도구의 사용이 증대되고 있는데, 분석/검증 도구로는 페트리 넷과 같은 수학적 형식론에 의한 방법과 시뮬레이션을 이용한 방법이 주로 사용되고 있다. 이 중에서 수학적 형식론은 분석 대상 시스템의 크기가 커지면 분석력에 한계점을 노출하고 있어서 산업계에서는 시뮬레이션 방법이 많이 사용되고 있는 실정이다.

그러나 기존의 시뮬레이션 방법은 제조 시스템의 레이아웃을 평가하거나 가동률, 시스템 체류 시간 등의 전체적인 성능 평가를 위해 주로 사용되고 있어서, 자동화 시스템 제어기 설계에 필요한 입/출력 시그널 단위의 상세 로직 표현 및 검증용으로는 한계점을 노출하고 있다.

그 이유로는 첫째, 상세 제어 로직을 표현하기 위해서는 제어기와 플랜트(plant)간의 상호작용을 모델링해야 하는데 기존의 시뮬레이션 방법은 작업물의 흐름(process view)이나 시스템 자원들의 서비스 행위(activity view)를 중심으로 한 모델링 방법을 채택하고 있기 때문에 구성 요소간 상호작용의 표현이 어려운 실정이다. 둘째, 최근의 자동화 제조 시스템들은 계층적이고 모듈러한 구조를 가지고 있고 표준화된 설비들을 조합하여 전체 시스템을 구축하는 추세이어서 이를 반영하기 위해서는 계층적 시스템 분해를 통한 타당한 접근 방법을 제공해야하는데 일반적인 시뮬레이션 방법들은 이러한 기능을 제공하고 있지 않고 있다.

위에서 설명한 자동화 제조 시스템 설계 및 시뮬레이션에 있어서의 새로운 요구사항을 반영하기 위해 최근에 객체지향 기술의 적용이 활발하게 연구되고 있는데, 일반적으로 객체지향 기술의 장점으로는 객체간 상호작용의 표현,

\* 책임저자(Corresponding Author)

논문접수 : 2006. 2. 9., 채택확정 : 2006. 6. 5.

한관희 : 경상대학교 산업시스템공학부/공학연구원

(hankh@gsnu.ac.kr)

※ 본 논문은 산업자원부 차세대신기술개발사업(차세대차량용 고안전경량 AI 차체부품개발)에서 지원하여 연구하였음.

계층적 구조, 재사용성, 수정 용이성 등을 들 수 있다[1].

본 연구에서는 자동화 제조 시스템- 특히, 자동 물류 핸들링 시스템의 기능적 요구사항을 분석하여 자동화 제조 시스템을 계층적이고 모듈러한 구성 요소들의 집합으로 정적 구조를 표현하고 제어기와 플랜트간의 상호작용으로 시스템의 동적 행태를 모델링하며, 이를 기반으로 시스템 성능 평가와 병렬적으로 제어기 로직을 검증할 수 있는 객체지향 설계 및 시뮬레이션 방법을 제안한다.

객체지향 기술과 제조 시스템 시뮬레이션 분야의 기존 연구들을 살펴보면, 우선 제조 시스템 분야에서 기존의 시뮬레이션 방법과 객체지향 시뮬레이션 방법을 비교하고 객체지향 시뮬레이션 도구들의 특징과 구조를 비교하는 연구가 수행되었다[2,3].

자동화 제조 시스템 모델링 연구로는 Choi[4]와 Park[5]의 연구에서 자동화 제조 시스템의 객체지향 시뮬레이션을 위해 레이아웃 모델, 기능 모델, 컨트롤 모델로 구성된 JR-Net (Job Resource relation Net) 모델링 프레임워크를 제안하였는데 제어 수준을 감독 제어(supervisory control) 레벨에 맞추었기 때문에 상세한 기기 레벨의 제어 모델을 제시하지는 않았다.

Calvo *et al.*에서는 자동화 시스템 설계에 객체지향 기술을 적용하는 방법을 제시하였으나 UML (Unified Modeling Language) 유즈 케이스 다이어그램이나 클래스 다이어그램을 이용하여 시스템의 정적인 구조를 표현하는데 그쳤다 [6]. Young *et al.*에서는 자동화 제조 시스템을 UML로 모델링하는 방법을 제시하고 그 결과를 PLC 코드로 변환하였으나 변환 방법을 제시하지는 않았다[7]. Bruccoleri and Diega에서도 UML을 이용하여 FMS 설계를 하고 이를 시뮬레이션으로 구현하였으나 감독 제어 레벨로 제어 수준을 국한하였다[8].

PLC 로직 설계 및 검증 도구에 관한 연구로 Spath & Osmer에서는 플랜트의 레이아웃 모델과 컨트롤 모델을 결합하여 시뮬레이션을 수행하고 이를 기반으로 PLC 코드를 생성하는 절차를 제시하였는데 구체적인 방법은 나타나 있지 않다[9]. Baresi *et al.*에서는 FBD (Function Block Diagram)을 이용하여 제어 로직을 설계하고 이를 페트리

넷으로 변환하고 SIMULINK 시스템을 이용하여 로직을 검증하고 그 결과를 C 언어로 생성하는 방법을 제시하였으나 FBD로 표현될 수 있는 간단한 로직으로 그 범위를 국한하였다[10].

본 연구에서 제시하는 로직 검증을 포함한 제어 로직 생성 절차는 (그림 1)에서와 같이 우선 자동화 제조 시스템 요구사항을 근거로 시스템 제어 로직을 위한 객체지향 분석/설계를 행한다. 그 후 산출된 설계 결과를 이용하여 시뮬레이션 모델을 생성하여 해당 모델을 실행함으로써 성능 평가와 병행하여 제어 로직의 타당성을 동시에 검증한다. 그 후 검증된 로직을 기반으로 PLC 코드를 작성하고 테스트한다. 테스트된 PLC 프로그램은 다른 구성 요소와 합쳐져서 통합 및 운영 테스트 단계를 거친다.

**II. 자동 물류 핸들링 시스템의 참조 모델과 객체지향 설계**

**1. 자동 물류 핸들링 시스템의 참조 모델**

최근 들어 물류 핸들링 시스템의 효율성이 CIM (Computer Integrated Manufacturing) 시스템의 구축 및 운영에 주된 성공 요인이라는 점이 강하게 인식되고 있다. 이 결과로 제조 시스템 설계자들 사이에 자동 물류 핸들링 시스템 설계를 지원하는 도구의 필요성이 점차 증대하고 있다. 자동 물류 핸들링 시스템은 자동화 제조 시스템의 한 종류로서 시작점에서 종착점에서의 작업물 이송을 주기능으로 하며 이를 위해 작업물의 분류, 집적, 로트 사이징 및 분배 등의 기능을 수행하고 작업물 막힘 현상이나 작업물 부족 현상이 없는 원활한 물류 제어를 그 목적으로 하고 있다.

자동 물류 핸들링 시스템을 구성하는 대표적인 설비들은 컨베이어(conveyor), AGV (Automatic Guided Vehicle), 스택커 크레인(stacker crane), 로봇(robot) 및 디버터(diverter) 등을 들 수 있으며 이송 목적과 이송 메커니즘에 따라 다양한 종류가 채택되기 때문에 사용 용도에 맞는 물류 핸들링 설비의 선정과 적합한 제어 로직 작성이 매우 중요한 요소가 되고 있다.

최근의 자동화 제조 시스템은 계층적이고 모듈러한 구조를 가지고 있고 표준화된 설비들을 조합함으로써 전체 시스템을 구축하는 추세이다. 즉, 유연성과 모듈화가 자동화 제조 시스템의 성공적인 운영에 매우 중요한 요소이므로 최근에는 제조 시스템 설계에 있어서 표준화된 설비를 사용하여 전체 시스템이 몇 개의 '스테이션(station)'으로 구성될 수 있도록 하는 모듈화 설계 개념을 채택하고 있다.

모듈화 설계를 반영하기 위해 일반적으로 자동화 제조 시스템을 분해해보면 (그림 2)의 클래스 다이어그램에서 보듯이 플랜트(plant)와 이를 제어하는 제어기(controller)와 플랜트/제어기의 운전을 담당하는 작업자(operator) 및 플랜트를 흐르는 작업물(workpiece)로 구성된다.

플랜트는 다시 기구부(mechanical parts), 감지지(sensor), 구동기(actuator) 및 MMI (Man Machine Interface)로 구성되며 (집합 연관 관계) 각 구성 요소는 고유의 속성(attribute)과 행위(method)를 가지고 있다.

그러므로 자동화 시스템 설계자는 설계시에 본 연구에서

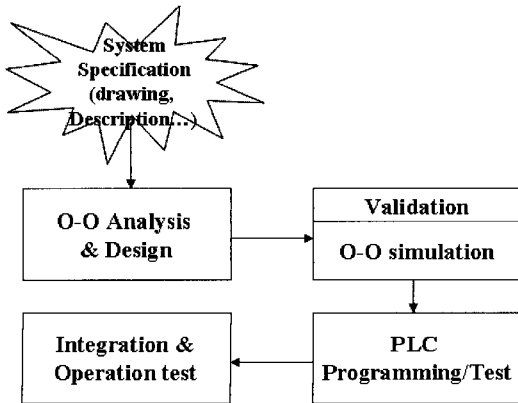


그림 1. 객체지향 제어 로직 생성 절차.

Fig. 1. Proposed control logic generation procedure.

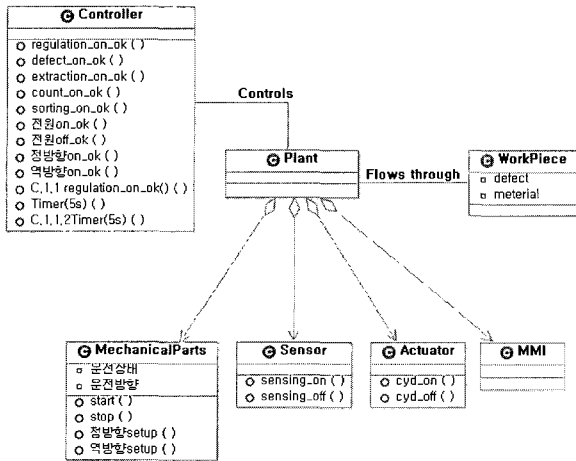


그림 2. 일반적인 자동화 제어 시스템 구조.  
Fig. 2. Generic AMS structure.

제시한 (그림 2)의 참조 모델을 기반으로 하여 대상 시스템의 특수한 요구사항을 반영하는 파생 클래스 다이어그램을 작성하여 원하는 시스템 구조 모델을 효율적으로 작성할 수 있다.

자동화 제조 시스템 중에서 자동 물류 핸들링 시스템은 작업물 형태와 이송 목적에 따라 다양한 종류의 설비 조합으로 구성된다. 이에 따라 물류 핸들링 설비는 이송 메커니즘에 따라 다양한 종류가 존재하는데, 본 연구에서는 이 중에서 산업 현장에서 많이 사용되는 컨베이어, 차량(vehicle), 디버터, 로봇에 대해 그 형태와 이송 메커니즘에 따라 상세하게 재분류한다.

비교적 균일한 크기나 무게를 가진 작업물을 고정된 경로를 통해 빈번하게 이송하는데 사용되는 컨베이어는 입력 지점에서 출력 지점으로 고정된 경로를 통해 작업물을 이송하는 직선형(linear)과 주어진 경로를 순회하는 순회형(circular)이 있다. 이 중에서 직선형은 연속적인 이송을 하는 연속형(continuous type)이 주를 이루고 있고, 순회형은 연속적인 이송을 하는 연속형과 정지/이동을 반복적으로 시행하는 단속형(intermittent type)으로 분류된다.

직선형 컨베이어는 다시 작업물 이송시의 집적 형태에 따라 작업물간 공간이 일정하게 유지되는 비집적형(non-accumulation)과 선행 작업물이 정지했을 때 후행 작업물들이 선행 작업물이 제거되거나 이동하기 전까지는 선행 작업물 앞으로 집적되는 집적형(accumulation)으로 다시 세분화된다.

디버터는 컨베이어 네트워크에서 물류 방향 전환용으로 사용되므로 컨베이어의 일부로 간주할 수 있으며, 로봇은 일정한 거리 내에서 작업물을 집어서 위치시키는(pick & place) 용도로 주로 사용된다.

비교적 자유로운 경로를 다른 작업물과 독립적으로 이송할 수 있는 차량형 타입은 정해진 주행 경로를 따라 주행하는 경로가이드형(guided path)과 주행 경로가 사전에 정해져 있지 않고 자유롭게 이동하는 자유경로형(free range)으로 구분된다. 이를 정리한 것이 (표 1)에 나타나 있다.

표 1. 물류 핸들링 설비 분류.

Table 1. Classification of material handling devices.

물류 핸들링 설비 분류				설비 예
컨베이어	직선형	연속형	집적형 비집적형	- ground conveyor
	순회형	연속형		- overhead trolley conveyor
		단속형		- power & free conveyor - carousel
차량	경로 가이드형			- AGV      - RGV - stacker crane
	자유 경로형			- truck      - fork lift
로봇				- robot
디버터				- diverter      - pusher - dropper

2. UML 기반 설계

UML (Unified Modeling Language)은 원래 소프트웨어 시스템의 분석 및 설계를 위한 객체지향 도구로 출발했으나 최근에는 기업의 업무 프로세스뿐만 아니라 자동화 시스템과 같은 실시간 시스템 설계에도 그 사용이 증가하고 있다. PLC 프로그램도 일종의 소프트웨어이므로 프로그래밍 전에 체계적인 분석 및 설계 과정이 필요하나 현행 PLC 프로그램 작성 과정은 이러한 부분을 결여하고 있다.

본 연구에서는 PLC 제어 로직의 체계적인 개발을 위해 UML에 의한 시스템 설계를 수행하고 이를 기반으로 시뮬레이션 모델을 작성하여 실행함으로써 제어 로직을 검증하는 절차를 제시한다. 객체지향 모델링 도구로는 IBM Rational Software Architect를 사용하였다[11].

본 연구에서 제시하는 PLC 제어 로직을 위한 시스템 설계는 1) 시스템의 기능적 요구사항을 표현하는 ‘기능 모델’과 2) 시스템의 정적 구조를 나타내는 ‘구조 모델’ 및 3) 시스템 구성 요소간의 상호작용을 나타내는 ‘상호작용 모델’의 3가지 모델로 구성된다.

‘기능 모델’은 유즈 케이스 다이어그램을 이용하여 시스템의 기능적 요구사항을 ‘use case’로 분류하여 모델링하는데, 자동화 시스템의 제어기(PLC)를 ‘system’으로 표현하고 감지기와 구동기 및 작업자(operator)는 ‘actor’로 표현한다. 자동화 시스템과 관련된 액터(actor)는 감지기와 구동기가 대부분이어서 감지기/구동기는 표준 심벌인 막대인간 대신에 UML 스테레오타입 기능을 이용하여 수정하여 사용하고 ‘시스템’ 경계의 왼쪽에는 PLC 입력부인 감지기를, 오른쪽에는 PLC 출력부인 구동기를 구분하여 표현한다.

그리고 분류된 각 유즈 케이스의 상세 사항은 유즈 케이스 기술서(use case description)에 의해 기술된다. 유즈 케이스 기술서는 하나의 기능 요구사항에 대해 필요한 선행/후행 조건을 명세화 하고, 그 기능을 수행하기 위해 필요한 시스템과 액터 사이의 상호작용을 사건 단위로 기술한 것이다. 유즈 케이스 다이어그램과 유즈 케이스 기술서의 예를 (그림 4)와 (그림 5)에 각각 나타낸다.

각각의 유즈 케이스를 실현하기 위해서는 문제 영역 내의 관련 클래스들이 상호작용에 의해 할당된 책임을 수행

하게 되며 시스템내의 연관 클래스들은 ‘구조 모델’에서 식별된다. 구조 모델은 (그림 2)와 같은 UML 클래스 다이어그램을 이용하여 작성한다.

그리고 유즈 케이스 기술서에서 표현된 시스템 수준에서의 상호작용은 ‘상호작용 모델’에서 순차적 처리가 구체화된다. 실제 자동화 제조 시스템은 제어기와 플랜트내 감지기/구동기 사이에서의 상호 시그널 송수신에 의해 작동되므로 이를 정밀하게 나타내기 위해서는 자동화 시스템내 구성 요소간의 상호작용을 상세하게 표현할 수 있어야 한다.

UML에서는 유즈 케이스를 실현을 위한 구성 요소간 상호작용 모델링 도구로 상태(state) 다이어그램, 순차(sequence) 다이어그램, 커뮤니케이션(communication) 다이어그램 및 활동(activity) 다이어그램 등을 제공하고 있다[12]. 이 중에서 자동화 시스템은 시간적인 순서 제약이 엄격하게 존재하므로 이를 모델링하기 위해서는 순차 다이어그램이 적합하다고 판단되어 본 연구에서는 순차 다이어그램을 이용하여 상호 작용 모델을 작성하였다.

순차 다이어그램에서는 특정 유즈 케이스를 실현하기 위해 관련된 객체간의 상호 작용을 시간의 흐름 순서로 기술하게 되는데 이 때 참여하는 객체들은 ‘구조 모델’에서 사전에 식별된다. 순차 다이어그램으로 표현된 ‘상호작용 모델’의 예는 (그림 7)에 나타나 있다.

**III. 연구 대상 시스템의 객체지향 설계**

이하에서는 자동 물류 핸들링 시스템의 일종인 컨베이어 시스템을 사례로 사용하여 본 연구에서 제시한 객체 지향 설계 절차를 설명한다.

**1. 대상 시스템 설명**

연구의 대상이 되는 시스템은 불량품의 식별/추출과 양품의 분류 (금속 제품/비금속 제품) 및 계수를 수행하는 컨베이어 기반 물류 핸들링 시스템으로 그 구조는 (그림 3)과 같다.

제품이 컨베이어 시스템 입구로 투입되면 투입 감지 센서에 의해 감지되어 우선 제품 간에 적절한 간격을 유지하기 위해 이송 간격 조절 실린더가 작동하여 일정 시간 (5 초) 동안 제품을 홀딩한다. 그 후 제품은 일정한 간격으로 컨베이어를 따라 흐르다가 양품/불량품을 식별하는 위치에 도착하면 2개의 광센서에 의해 불량품을 식별한다.

양품/불량품은 제품의 높이에 따라 구분하는데 정해진 규격보다 낮거나 높은 제품은 불량품이 된다. 즉, 동일한 지점에 설치되어 있는 ‘높은 위치 센서’와 ‘낮은 위치 센서’ 중에서 ‘낮은 위치 센서’만 감지된 제품이 양품으로 판정된다. 나머지 경우는 불량으로 판정된다.

불량으로 판정된 제품은 공압 실린더에 의해 추출되며 양품은 금속 감지 근접 스위치에 의해 금속/비금속 여부가 판별된다. 금속인 경우에는 공압 로터리 실린더가 회전하여 제품을 컨베이어의 진행 방향 좌측으로 유도한다. 비금속인 경우에는 컨베이어 진행 방향 우측으로 흐른다. 그 후 컨베이어 끝단에 설치된 광센서에 의해 양품 수량이 계수되며 누적 생산 수량이 LED 디스플레이에 나타난다.

시스템의 작동을 위해 감지기로는 5개의 광센서와 2개의

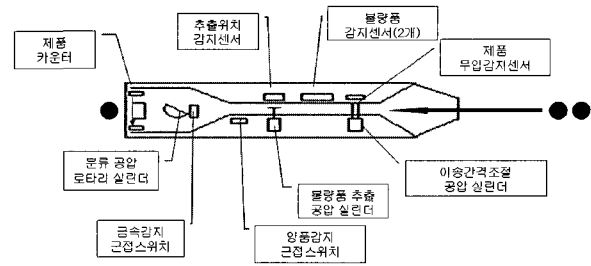


그림 3. 컨베이어 기반 물류 핸들링 시스템 구조.  
Fig. 3. Structure of conveyor-based material handling system.

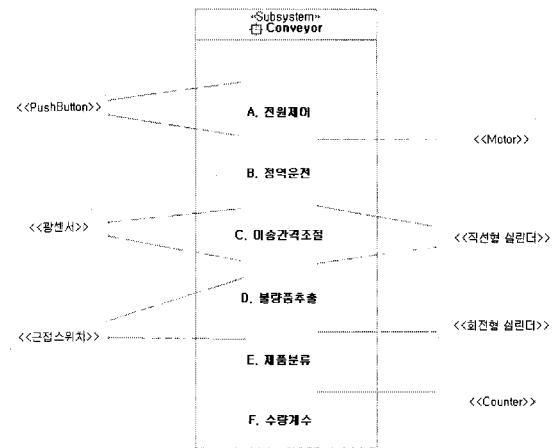


그림 4. 유즈 케이스 다이어그램.  
Fig. 4. Use case diagram.

근접 스위치가 사용되며 구동기로는 2개의 모터와 2개의 직선형 실린더 및 1개의 로터리 실린더가 사용되고 MMI (Man Machine Interface)로는 하나의 카운터 디스플레이가 사용된다.

**2. UML 설계**

우선 대상 시스템의 기능적 요구사항을 분석하기 위해 유즈 케이스 모델을 작성하였다. 불량품 추출 및 양품 분류를 목적으로 하는 대상 시스템의 기능은 (그림 4)에 나타난 것과 같이 크게 6가지 기능으로 분류할 수 있다 (전원 제어, 정역 운전, 이송 간격 조절, 불량품 식별 및 추출, 제품 분류, 수량 계수).

제어기를 나타내고 있는 ‘subsystem’의 왼쪽에는 관련 액터인 푸시버튼, 광센서, 근접 스위치와 같은 감지기들이 배치되고 오른쪽에는 모터, 실린더, 카운터 등의 구동기 액터들이 배치된다. 이와 같이 자동화 제조 시스템을 수행 기능으로 분해하는 것은 추후에 시스템을 모듈 단위로 구현하기 위해 필요한 방법이다.

유즈 케이스 다이어그램에 의해 기능적 요구사항이 파악된 후, 각 유즈 케이스에 대해 유즈 케이스 기술서를 작성한다. 유즈 케이스 기술서에는 단위 유즈 케이스에 대한 시나리오, 선행/후행 조건 및 시스템과 액터 사이의 상호작용을 이벤트 단위로 표현하게 된다. (그림 5)에 대상 시스템의 유즈 케이스 중에서 ‘C. 이송 간격 조절’ 유즈 케이스에

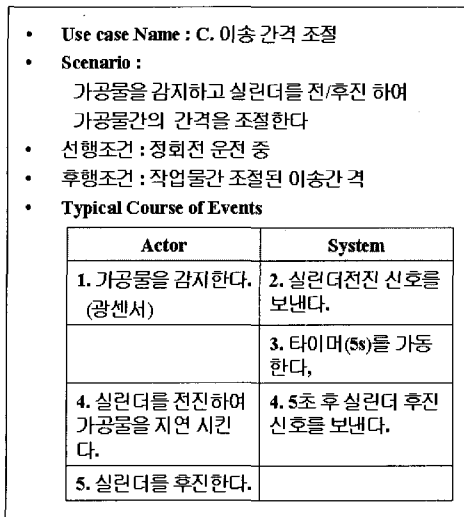


그림 5. 유즈 케이스 기술서.  
Fig. 5. Use case description.

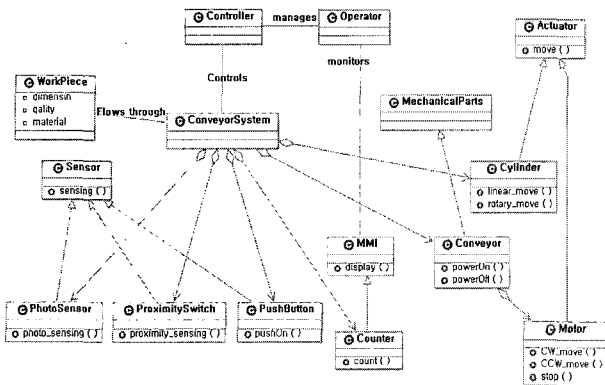


그림 6. 클래스 다이어그램.  
Fig. 6. Class diagram.

대한 기술서를 나타내고 있다.

‘구조 모델’을 위해서는 클래스 다이어그램을 작성하는데, (그림 6)는 (그림 2)에서 제시된 표준적인 자동화 제조 시스템의 구성 요소들을 표현하는 참조 모델에 근거하여 연구 대상인 컨베이어 기반 물류 핸들링 시스템을 표현한 것이다.

(그림 6)에서 보듯이 컨베이어 기반 물류 핸들링 시스템은 (그림 2)에서 제시된 참조 모델의 서브 모델로 표현할 수 있으며, 감지기, 구동기, 구조부 및 MMI 클래스는 대상 시스템의 세부적인 특성에 따라 공통적인 속성과 메서드들을 공유하는 복수 개의 서브클래스로 나뉜다 (상속 관계).

예를 들면 대상 시스템에서 사용되는 감지기인 광센서, 근접 스위치 및 푸시 버튼은 플랜트 클래스인 ‘conveyor system’을 구성하는 요소이면서 공통적인 특성은 감지기 클래스에서 상속받는 구조를 취하고 있다. 각 클래스들은 그 클래스가 수행하는 구체적인 행위(메서드)를 명세하게 되는데 이는 상호작용 모델에서 구체화 된다

‘상호작용 모델’에서는 ‘기능 모델’의 유즈 케이스 기술서와 ‘구조 모델’의 클래스 다이어그램을 기반으로 순차 다

이어그램을 작성한다. 상호작용 모델의 일부로서 (그림 4)의 유즈 케이스 D에 해당하는 ‘불량품 식별 및 추출’ 기능을 위해 각 객체들이 시간에 따라 어떤 메시지를 주고받는가를 표현한 순차 다이어그램을 (그림 7)에 나타내는데 제어기(Controller)를 기준으로 왼쪽에 감지기 객체가 오른쪽에 구동기 객체가 배치되어 있다.

즉, 불량품 식별 및 추출을 위해 제어기와 높은 위치/낮은 위치 감지 광센서, 추출 위치 감지 광센서, 양품 감지 근접 스위치 및 불량품 추출용 직선형 실린더 사이에서 발생하는 시간의 흐름에 따른 메시지 교환을 표현하고 있다. 이 순차 다이어그램에 근거하여 각 객체가 수행해야할 메서드들의 종류와 내부 로직이 상세화 된다.

**IV. 시뮬레이션 모델 생성 및 실행**

**1. 시뮬레이션 모델 생성 절차**

III장에서 작성된 3가지 설계 모델을 이용하여 객체지향 시뮬레이션 모델을 작성하게 되는데, 본 연구에서는 시뮬레이션 도구로 emPLANT[13]를 이용하였다.

우선 ‘기능 모델’의 유즈 케이스 다이어그램을 이용하여 대상 시스템의 기능적 요구사항을 파악하고 유즈 케이스 기술서를 이용하여 시스템 레벨에서의 제어기와 감지기/구동기 사이의 상호작용을 파악한다. 그 후 ‘구조 모델’에서 파악된 시스템 구성 요소의 계층 구조와 구성 요소 간의 상속 및 연관 관계를 이용하여 각 구성 요소들을 시뮬레이션 모델 구성 요소로 변환한다.

(그림 2)에서 표현된 일반적인 자동화 제조 시스템 구성 요소와 emPLANT 시뮬레이션 요소와의 매핑은 (표 2)과 같다.

시스템의 정적 구조가 확정되면 각 유즈 케이스를 실현하기 위해 관련된 구성 요소 간의 메서드 호출에 의한 상호 작용이 필요한데 이는 순차 다이어그램으로 표현된 ‘상호작용 모델’을 이용하여 식별한다.

순차 다이어그램에서 식별된 각 메서드들이 시뮬레이션 수행시 실행하는 구체적인 로직은 emPLANT 자체 언어인 SimTalk를 이용하여 프로그래밍 한다.

예를 들면 (그림 7)의 D.3.1 메서드인 extractOn\_ok() 메서드는 불량품 추출 위치 광센서가 제품을 감지하여 제어기에 ‘ON’ 신호를 보내면 제어기가 수행하는 메서드인데, 그 내용은 제품이 양품인지 불량인지 판단하여 불량일 경

표 2. UML 설계 모델과 시뮬레이션 구성 요소간 매핑.

Table2. Mapping between UML elements and simulation elements.

Generic AMS class		emPLNAT class
controller		frame/method
workpiece		entity
plant	sensor	SingleProc/Line-sensor
	actuator	SingleProc
	mechanical parts	Line/SingleProc/Transporter
	MMI	Frame/Method

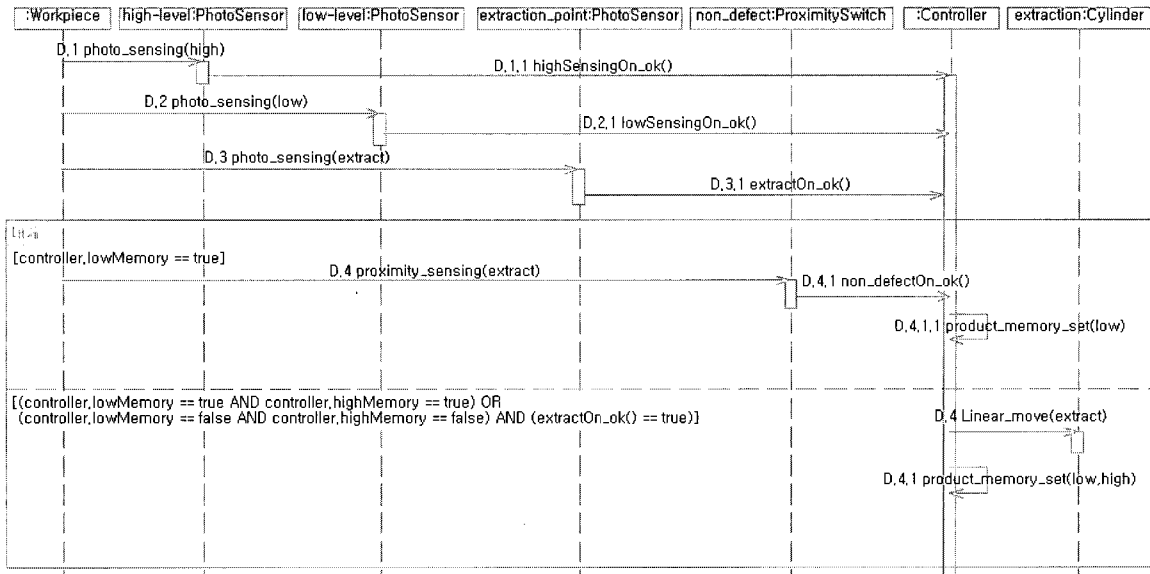


그림 7. 순차 다이어그램.  
Fig. 7. Sequence diagram.

우 구동기인 추출 실린더를 작동시키는 것(linear\_move (extract) 메서드 실행)으로 (그림 8)에 SimTalk로 구현된 코드를 나타낸다.

2. 대상 시스템의 시뮬레이션 모델 생성 및 실행

객체지향 모델은 그 특성상 사전에 정의된 클래스를 상속받아 공통 특성을 공유하는 확장된 새로운 클래스를 만들 수 있고 하나의 시스템을 계층적으로 분해할 수 있어서 타당한 방식으로 모델링을 수행하기 용이하다.

연구 대상 시스템은 2계층으로 구성되었는데, 상위 계층 모델은 (그림 9)의 오른쪽 상단과 같이 시스템 구성 요소가 PLC로 대표되는 제어기, 플랜트, 입력 원천(source), 불량품 출력(defect\_storage) 및 양품 출력(good\_storage)만으로 구성된다. 즉, 전체 시스템을 플랜트와 제어기로 간략화하여 표현할 수 있다. 하위 계층 모델은 상위 계층 모델을 분해하여 상세화한 것으로 (그림 9)의 하단에서와 같이 감지기나

구동기와 같은 구체적인 구성 요소들이 배치되어 실제적인 시뮬레이션을 실행하는 기본 모델이 된다.

시뮬레이션 모델과 실험 계획이 작성되고 난 후, (그림 9)의 하단 시뮬레이션 모델이 실행되는데, 이 모델에서는 작업물의 흐름이 애니메이션으로 나타나며 이러한 작업물의 흐름에 따른 감지기의 감지 사건이 PLC로 전달되면서 PLC의 작동 상황을 작업물의 흐름과 병렬적으로 확인할 수 있다.

즉, 시뮬레이션 실행시 컨베이어에 설치된 감지기에서의 작업물 감지는 (그림 9)의 왼쪽 화면에 있는 PLC 입력 포트에 'ON' 신호를 보내게 되며 PLC는 이에 반응하여 (그림 8)과 같은 해당 로직을 실행하여 특정한 출력 포트에 'ON' 신호를 보내게 된다. 출력된 'ON' 신호는 구동기에 전달되어 구동기를 작동시킨다. 이와 같이 작업물의 흐름에 따라 PLC와 감지기/구동기 사이의 시그널 입출력이 (그림 9) 왼쪽에 표현된 PLC의 해당 포트에 'ON/OFF'로 애니메이션 되면서 나타나게 된다. 여기에서 PLC의 왼쪽 부분은 입력 포트를, 오른쪽 부분은 출력 포트를 나타내며 'ON' 신호는 빨간 색으로 표시된다.

PLC 프로그램 개발자는 이와 같이 물류 흐름과 PLC 입출력 포트 시뮬레이션 과정을 동시에 병렬적으로 체크하면서 설계된 PLC 로직의 작동 모습과 오류 가능성을 사전에 검증할 수 있다. 즉, 객체지향 설계 및 시뮬레이션을 이용하여 기존의 이산 사건 시뮬레이션 시스템의 주된 용도인 시스템 성능 평가와 더불어 자동화 제조 시스템을 제어하는 제어기 (PLC)의 로직 검증을 병렬적으로 수행할 수 있게 된다.

위에서와 같이 시뮬레이션 실행에 의해 제어 로직이 검증된 후 이를 기반으로 PLC 프로그램이 작성되는데, 프로그램 작성을 위해서는 1) '상호작용 모델'에서 작성한 순차 다이어그램을 이용하여 제어 로직의 실행 순서를 파악하고,

```

.Models.PLC.extractOn_ok
{
  is
  do
    if (highLevel_Value=1 and
        lowLevel_Value=1) or (highLevel_Value
                              =0 and lowLevel_value=0)
    then
      .models.conveyor_system.Plant.linear_move_extract;
      product_memort_set_lowhigh;
    end;
  end;
}
    
```

그림 8. 심토크 시뮬레이션 코드 예.  
Fig. 8. SimTalk code example of controller method.

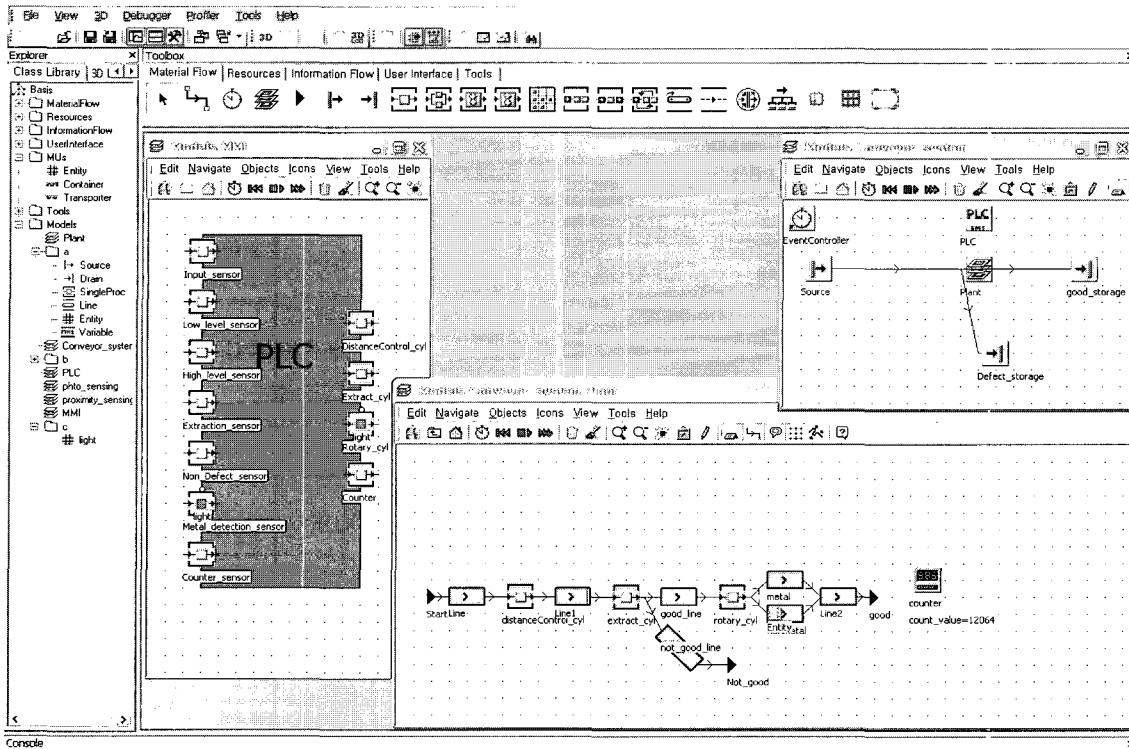


그림 9. 객체지향 시뮬레이션 모델.  
 Fig. 9. Object-oriented simulation model.

2) 시뮬레이션 모델 생성시 SimTalk로 작성된 각 메시지를 이용하여 실행 로직을 파악한다.  
 (그림 10)은 (그림 4)의 'D. 불량품 식별 및 추출' 유즈케이스를 래더 다이어그램으로 작성한 예이다. (그림 10)의 행4-5는 (그림 8)의 코드를 구현한 것이다. PLC 프로그래밍 소프트웨어로는 GMWIN을 사용하였다[14].

**V. 결론**

본 연구에서는 점차 복잡해져가는 자동화 제조 시스템의 제어 로직을 실제 시스템 설치 전에 검증하여 오류 가능성을 제거하기위해 여러 분야의 엔지니어들이 모두 이해 할 수 있는 객체지향 설계 및 검증 시뮬레이션 방법을 제안하였다. 즉, 자동화 제조 시스템의 객체지향 설계와 시뮬레이션을 통해 기존의 이산 사건 시뮬레이션 시스템의 주 용도인 성능 평가와 병행하여 제어기 상세 로직을 검증할 수 있음을 보였다. 특히, 객체지향 설계는 최근의 자동화 제조 시스템 추세인 계층적 모듈화와 표준화된 설비의 사용을 시스템 설계에 효과적으로 반영할 수 있는 방법으로 판단 된다.

이와 같이 객체지향 시뮬레이션을 이용하여 자동화 시스템의 설계 단계에서부터 제어기의 상세 로직을 검증해 봄으로써 실제 시운전 단계에 가서야 발견되는 여러 가지 로직 오류를 사전에 제거할 수 있게 된다. 그리고 생성된 시뮬레이션 모델은 자동화 제조 시스템과 관련된 여러 분야의 엔지니어들이 공통적으로 활용할 수 있는 의사소통 도구로서도 주요한 역할을 할 수 있다.

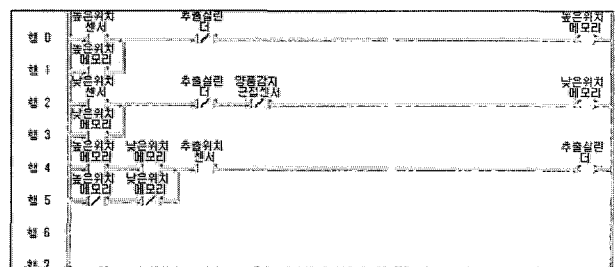


그림 10. 래더 다이어그램.  
 Fig. 10. Ladder diagram.

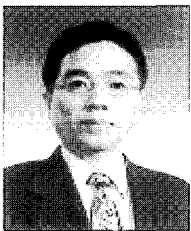
추가 연구로는 시뮬레이션 결과를 이용하여 이를 PLC 래더 다이어그램으로 자동 변환하는 연구와 자동화 제조 시스템의 시뮬레이션 모델을 생성하기 위한 체계적인 객체지향 설계 방법에 관한 연구가 필요하다.

또, 본 연구에서 제시한 방법을 자동차 라인과 같은 대규모 시스템에 적용하기 위해서는 시스템을 체계적으로 하향 분할하여 분석 및 설계를 행하고 이를 합성하여 시뮬레이션 모델을 실행함으로써 문제의 크기를 줄이는 계층적 시스템 분해를 통한 하향식 설계 방법론에 관한 연구도 시급히 요구된다.

**참고문헌**

[1] D. F. Geuder "Object-oriented modeling with simple++," *Proceedings of the 1995 Winter Simulation Conference*, Arlington, VA, USA, 1995.

- [2] S. Narayanan, D. A. Bonder, U. Srekanth, T. Govindaraj, L. F. McGinnes, and C. M. Mitchell, "Research in object-oriented manufacturing simulations: an assessment of the state of the art," *IIE Transactions*, vol. 30, no. 9, pp. 795-810, 1998.
- [3] P. Ball and D. Love, "Expanding the capabilities of manufacturing simulators through application of object-oriented principles," *Journal of Manufacturing Systems*, vol. 13, no. 6, pp. 412-423, 1994.
- [4] B. K. Choi, K. H. Han, and T. Y. Park, "Object-oriented graphical modeling of FMSs," *The International Journal of Flexible Manufacturing Systems*, vol. 8, no. 2, pp. 159-182, 1996.
- [5] T. Y. Park, K. H. Han, and B. K. Choi, "An object-oriented modeling framework for automated manufacturing system," *International Journal of Computer Integrated Manufacturing*, vol. 10, no. 5, pp. 324-333, 1997.
- [6] I. Calvo, M. Marcos, D. Orive, and I. Sarachaga, "Using object-oriented technologies in factory automation," *Proceedings of the 2002 IECON (Industrial Electronic Conference)*, pp. 2892-2897, Sevilla, Spain, 2002.
- [7] K. W. Young, R. Piggan, and P. Rachitrangan, "An object-oriented approach to an agile manufacturing control system design," *International Journal of Advanced Manufacturing Technology*, vol. 17, no. 11, pp. 850-859, 2001.
- [8] M. Bruccoleri and S. N. Diega, "An object-oriented approach for flexible manufacturing control systems analysis and design using the unified modeling language," *The International Journal of Flexible Manufacturing Systems*, vol. 15, no. 3, pp. 195-216, 2003.
- [9] D. Spath and U. Osmer, "Virtual reality-an approach to improve the generation of fault free software for programmable logic controllers (PLC)," *Proceedings of the IEEE International Conference on Engineering of Complex Computer Systems (ICECCS)*, pp. 43-46, Montreal, Canada, 1996.
- [10] L. Baresi, M. Mauri, A. Monti, and M. Pezze, "PLCTools: Design, formal validation, and code generation for programmable controllers," *Proceedings of 2000 IEEE Conference on Systems, Man and Cybernetics*, Nashville, USA., 2000.
- [11] IBM, Rational Software Architect, <http://www.306.ibm.com/software/awdtools/architect/swarchitect/index.html>, 2006.
- [12] H.-E. Eriksson, M. Penker, B. Lyons, and D. Fado, *UML 2 Toolkit*, Wiley Publishing, Indianapolis, Indiana, 2004.
- [13] Unigraphics, eM-Plant, [www.ugs.com/products/tecnomatix/plant\\_design/em\\_plant.shtml](http://www.ugs.com/products/tecnomatix/plant_design/em_plant.shtml), 2006.
- [14] LS Industrial Systems, *GMWIN V4 User Manual*, 2006.



#### 한 관 희

1982년 아주대학교 산업공학과 졸업.  
1984년 KAIST 산업공학과 석사. 1996  
년 KAIST 자동화 및 설계공학과 박사.  
2000년~현재 경상대학교 산업시스템공  
학부 부교수. 관심분야는 객체지향 모  
델링.