

# 제조 셀 모델 플랜트의 운영 소프트웨어의 개발

이준수 · 김종근 · 정병호<sup>†</sup>

전북대학교 산업시스템공학과

## Development of an Operating Software for the Manufacturing Cell of a Model Plant

Jun-su, Lee · Jong-keun Kim · Byung-Ho Jeong

Department of Industrial and System engineering, Chonbuk National University, Chonbuk, 561-756

This paper developed an operating software for the manufacturing cell of a model plant. The model plant, which was manufactured by Fishertechnik, consists of an assembly cell, a machining cell, and supplementary material handling equipments. Each I/O module of the model plant communicates with the INTERBUS controller devices via High-level Language Interface(HLI). The machining center has two machining cells, which consists of 4 machines respectively, two 3-axis portal transporters, and a rail guided vehicle(RGV). The Petri-Net was used to investigate the dynamics of each machining cell, e.g., the relation of material handling equipments and machines. The operating software was analyzed and designed by Object Oriented Technique. The software was implemented using Delphi 3.0 under Windows 95/NT operating system.

**Keywords:** CIM, operating software, INTERBUS

### 1. 서론

시장 환경이 소비자 중심으로 변화함에 따라 기업의 생산 환경은 다품종 소량 생산체제로 변화하고 있으며 이를 뒷받침해 줄 유연성을 가진 생산 시스템의 구축에 대한 필요성이 끊임 없이 대두되고 있다(S.K. Son *et al.*, 1991). 생산의 유연성과 효율성을 추구하는 유연생산시스템(flexible manufacturing system, FMS)과 컴퓨터통합생산(computer integrated manufacturing, CIM) 시스템 도입에 많은 업체들이 관심을 가져왔다. CIM 환경 하에서 기본적인 생산 활동을 통합하는 샵플로어 제어시스템(shop floor controller system, SFCS)의 역할은 매우 중요하다. SFCS의 주요 기능으로는 자동 장비 통제와 다른 생산 활동의 운영을 조정하고 감시(monitor)하는 역할을 들 수 있다. CIM 시스템에서 자원 할당, 공정 선정, 작업 지시, 시스템 상태의 보고, 시

스템 에러 감지 및 복구, 공정 상태 모니터링 등 다양한 운영 요소들을 통합하는데 많은 어려움이 존재하므로 위의 요소들을 만족시키기 위한 SFCS의 설계 및 개발은 더욱더 복잡한 시스템으로 발전하게 된다(Lie *et al.*, 1998).

Jones and Mclean(1986)은 퍼실리티 레벨(facility level), 샵 레벨(shop level), 셀 레벨(cell level), 작업장 레벨(workstation level), 장비 레벨(equipment level)의 5레벨로 이루어진 자동 생산 시스템(automated manufacturing system, AMS)의 계층적 구조를 제시하였다. O'Grady(1987)는 자동 생산 시스템을 공장(factory) 레벨, 샵 레벨, 셀 레벨, 장비 레벨의 4레벨로 분류했다. 공장 레벨에서는 비교적 긴 기간의 생산 시스템에 관한 전략 및 목적 등을 샵 레벨로 전달한다. 샵 레벨에서는 공장 레벨에서 받은 정보를 셀 레벨에서 생산할 정보나 신호 등으로 변환한다. 셀 레벨에서는 장비 레벨에서 장비 운영을 실행하도록 명령을 준다.

이 논문은 공업기술연구소의 지원에 의해 이루어졌습니다.

<sup>†</sup>연락처: 정병호 교수, 561-756 전북 전주시 덕진구 덕진동1가 756번지 전북대학교 산업시스템공학과, Fax : 063-270-2333

e-mail : jeong@moak.chonbuk.ac.kr

2002년 1월 접수, 2회 수정 후 2002년 10월 게재 확정.

표 1. CIM 계층적 구조에 관한 기존 연구

기본기능		기존연구	NIST-AMF Model <sup>1)</sup> (1990)	O'Grady (1987)	Jones & Mclean (1986)	Smith & Joshi <sup>2)</sup> (1995)
제품 설계와 생산 기술, 구매 등	공장 (Facility / Factory)	공장 레벨	팩토리 레벨	퍼실리티 레벨		
생산 관리 등	에어리어(Area)	샵플로어 레벨	샵레벨	샵레벨	샵레벨	
제조(공정 레벨)	셀(Cell)	셀 레벨	셀 레벨	셀 레벨	셀 레벨	
제조(스테이션 레벨)	스테이션(Station)	작업장 레벨		작업장 레벨	작업장 레벨	
제조(장비 레벨)	장치(Equipment)	장비 레벨	장비 레벨	장비 레벨	장비 레벨	

註: 1) NIST-National Institute Standard and Technology

2) 샵플로어 제어기 개발

Jones and Saleh(1990)는 생산 시스템의 복잡성을 다루기 위해 플랜트(plant) 레벨, 샵플로어(shop floor) 레벨, 셀 레벨, 장비 레벨의 계층 구조로 이루어진 생산 시스템인 NIST 모델을 제안하였다. Smith and Joshi(1995)는 자동 생산 환경을 위한 샵플로어 제어기를 개발하는 데 있어 객체지향 접근 방법 및 클래스 등을 제시하였다. 제시된 제어 구조는 일반화된 3레벨, 즉 셀 레벨, 작업장 레벨, 장비 레벨로 분류하였다. Liu *et al.* (1998)은 샵플로어 제어시스템을 분석하기 위해 객체지향 분석 및 설계 기법을 이용하여 프로그램의 모듈화 및 재사용성(reusability)을 추구하였다. 이상의 연구 결과들은 <표 1>에서 요약하고 있다.

본 논문은 모델 플랜트를 이용한 교육용 CIM 시스템 구축의 전(前) 단계로서 샵 레벨에서의 운영 소프트웨어를 개발하기 위한 것이다. 샵 레벨에서는 생산 현장에서의 제조 활동 및 각종 제조 지원 활동을 통제한다. 조영호와 정병호(1999)는 모델 플랜트의 자동 창고와 조립 라인에 대한 운영 소프트웨어 구현에 있어서 DDI(Device Driver Interface)를 사용하였다. 본 논문에서는 모델 플랜트의 제조 셀을 중심으로 자동 창고와 운반 장비를 포함한 샵 레벨과 셀 레벨을 운영하는 소프트웨어를 개발함으로써 모델 플랜트 전체의 운영 소프트웨어를 완성하였다. 샵 레벨 제어를 위해 기존의 INTERBUS의 DDI보다 상위 개념인 HLI (High-level Language Interface)를 사용하여 프로세서를 제어하였다.

시스템의 정적 구조 및 동적 사건들을 모델링하기 위하여 Petri-Net 기법을 이용하여 전체 공정의 흐름을 분석하였으며, 실제 운영 소프트웨어를 구축하기 위한 방법으로 객체지향 기법을 사용하였다. 이는 차후에 개발될 모델 플랜트를 대상으로 교육용 CIM 시스템의 공장 레벨을 완성하기 위한 기반이 될 것이다.

2절에서 모델 플랜트와 INTERBUS 컨트롤러에 대해 간략하게 소개한다. 3절에서는 가공 셀의 동적 특성에 대한 Petri-Net 모델을 보이고, 4절에서는 객체지향 설계를 이용한 운영 소프트웨어 설계, 5절에서는 제조 셀의 운영 소프트웨어의 개발에 대하여 서술한다.

## 2. 모델 플랜트와 교육용 CIM

### 2.1 모델 플랜트

본 연구에 사용된 시스템은 CIM에 대한 교육 및 연구용으로 도입한 시스템으로 독일의 Fischertechnik사에서 제작하였으며, 모델 플랜트는 자동 창고와 가공 및 조립 라인으로 구성되어 있으며, 각 모듈은 단품들의 결합에 의해 이루어져 있다.

<그림 1>은 모델 플랜트의 전체 배치도이다. 점선 부분이 본 연구의 대상 모듈로 2개의 머시닝 셀(machining cell, MC)과 셀 사이를 움직이면서 부품의 적재/하역을 담당하는 RGV(rail guided vehicle)로 구성된다. 각 셀(cell1, cell2)은 4개의 머신 테이블(machine table, MT)과 셀내 이동 및 셀과 RGV 간의 작업물 이동을 처리하는 3-portal(3-axis portal transporter)로 구성된다. 그의 자동 창고 및 조립 라인 모듈이 있고, INTERBUS-PC 인터페이스 보드에는 어셈블리 언어(assembly language)나 상위 레벨 언어(high-level language)를 이용하여 제어할 수 있도록 펌웨어가 장착되어 있다. <표 2>는 가공 부품들과 ball, disc, cover, box로 이루어진 완성품을 보여준다. 가공 부품에 대한 가공 경로 및 가공 시간은 <표 3>과 같다.

제조 셀 내의 작업물의 흐름을 살펴보면, 자동 창고에서 가공할 원재료를 스택커 크레인이 I/O\_#2로 적재하여 CB\_#4를 지나 CB\_#5에 도착한다. 여기에서 RGV에 적재되고 이동할 MC에 따라 3-portal\_#1 이나 3-portal\_#2에 적재되어 작업할 MT에 도착하게 된다. 특정 MT에서 작업을 마친 job은 3-portal

표 2. 가공 품목

종류	품번					비고
	P1111	P1122	P2213	P2114	P3123	
완성품						조립
BALL	A4001	A4002	A4001	A4001	A4002	가공
DISC	A4003	A4004	A4004	A4003	A4004	가공
COVER	A1001	A2001	A3001	A2001	A3001	가공
BOX	B1001	B1002	B1003	B1002	B1003	가공

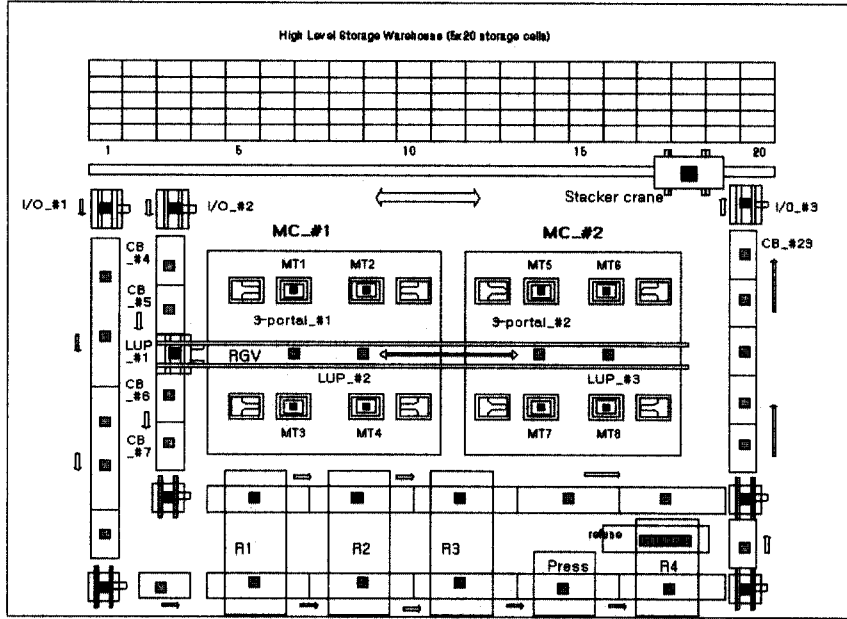


그림 1. 모델 플랜트 배치도.

표 3. 셀에서의 부품별 가공 시간

품 번	원재료	셀#1		셀#2		가공시간		총가공 시간	
		MT	MT	MT	MT	PT	PT		
A1001	R1001	1	2			1	1.1	2.1	
A2001	R2001	2	3	1		1.2	1.3	3.5	
A3001	R3001	1	4			1	1.2	2.2	
A4001	R4001	1				1.2		1.2	
A4002	R4002	3	2	1		1.5	1.1	3.6	
A4003	R4003	1	3			1.8	1.2	3	
A4004	R4004	1	2			1.1	1	2.1	
B1001	R0001				6	7	2.3	2.5	4.8
B1002	R0002				5	7	2.7	2.5	5.2
B1003	R0003				6	8	2.3	2.8	5.1

에 적재되어 다음 작업할 MT로 이동하거나 가공 작업을 완료된 완제품은 RGV를 경유하여 MC에서 나오게 된다. 이런 과정을 거쳐 조립될 부품이 완성된다. 이 모델 플랜트의 MC에서는 실제 가공은 이루어지지 않고 적재(loading)된 상태에서 지정된 가공 특성에 따라 가공 시간만큼 시간을 지연 후 방출하는 것으로 가공을 대신한다.

2.2 INTERBUS 컨트롤러와 통신

필드 버스(field bus)는 자동화 기기에서 생성되는 데이터를 실시간으로 처리하며 첨단 생산자동화 및 분산 제어시스템의 네트워크 구조상 가장 기본이 되는 네트워크이다. 필드 버스의 표준을 만들기 위하여 노력한 결과 현재는 Profibus, INTERBUS, Bitbus, Controlnet, Fieldbus Foundation(FF), P-net 등이 세계 표준(IEC 61158)으로 제정되어 사용자가 현장 상황에 알맞게 선택하여 사용할 수 있도록 복합표준으로 제정되었다.

본 연구에서 사용된 INTERBUS는 Phoenix Contact사에 의해 개발되어 센서/액츄에이터 레벨로부터 생산 자동화, 공정 자동화 분야에서 널리 사용되는 산업용 솔루션이다(INTERBUS User manual, 1997).

컴퓨터 통신망이 완전한 기능을 수행하기 위해서는 OSI(open system interconnection)의 7계층 구조를 모두 갖추어야 하지만, 필드 버스는 앞서 언급한 실시간 동작에 대한 요구 사항을 만족시키기 위하여 <그림 2>와 같이 물리적 계층(physical layer)과 데이터 링크 계층(data link layer) 위에 바로 응용 계층(application layer)이 위치하고, 그 위에 사용자 계층(user layer)이 위치하는 4계층의 구조를 가진다(Glanzer et al. 1996).

본 연구에서 사용된 응용 프로그램과 모델 플랜트 사이의 통신 인터페이스는 HLI로 INTERBUS 사용자에게 간편하고 효율적인 프로그래밍 인터페이스이다(INTERBUS User manual 1998).

HLI의 장점으로는 운영시스템과 독립적인 하드웨어는 INTERBUS에 접근할 수 있고, 변수명을 사용하여 쉽고 신속하게 데이터 교환, 모든 컨트롤러 보드에 동일하게 접근이 가능하다. HLI는 기존의 DDI를 기반으로 운영되며 구조는 <그림

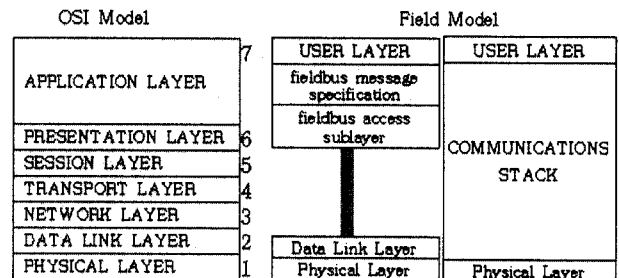


그림 2. OSI 및 필드 버스 모델 구조.

표 4. HLI 함수 및 역할

HLI 함수	역 할
IBS_HLI_Init	제어반 초기화
IBS_HLI_Exit	제어반 마침
IBS_HLI_GetControllerInfo	제어반 정보요청
IBS_HLI_GetBusInfo	버스환경에 관한 정보
IBS_HLI_Process	상태기계의 처리
IBS_HLI_PD_Input	입력 PDO 갱신
IBS_HLI_PD_Output	출력PDO 이송
IBS_HLI_ActivateWatchdog	watchdog 활성화
IBS_HLI_StartBus	INTERBUS 시작
IBS_HLI_StopBus	INTERBUS 정지
IBS_HLI_AlarmStop	INTERBUS 경고 멈춤
IBS_HLI_RunCycle	데이터변환 사이클 시작
IBS_HLI_RunCompleteCycle	INTERBUS 사이클 실행

그림 3. HLI 인터페이스와 INTERBUS 드라이버 구조.

3>과 같고 HLI 함수 및 역할은 <표 4>에 정리되었다. HLI는 INTERBUS 함수에 지능적인 사용자 인터페이스를 제공하며 응용 프로그램의 요구들을 INTERBUS의 컨트롤러 보드에서 복잡한 통신 절차를 변환하는 역할을 한다.

2.3 교육용 CIM

최종적으로 개발하고자 하는 시스템의 개념적 구조는 <그림 4>와 같다. 플랜트 레벨 제어기에는 MRP 모듈, 스케줄링 모듈, 자재관리 모듈, Dispatching 모듈을 포함하고 있다. MRP 모듈에서 주일정계획(MPS)이 수립되고, 부품의 BOM 정보를 이용하여 BOM이 전개되고, 스케줄링 모듈에서는 작업장별 세부 일정계획이 수립된다. Dispatching 모듈은 스케줄링 모듈로부터 수립된 세부 일정을 제어기 모듈에 작업을 넘겨주는 역

할을 한다. 그러면 제어기 모듈은 하위 cell에 작업을 지시한다. 자동 창고(AS/RS) 제어, 조립 제어, 가공 제어, 반송 제어를 거쳐 원재료, 반제품 등이 가공 또는 조립되는 생산 과정을 거친다. 조립, 가공 등에 의한 현황 정보의 변화는 DB에 반영됨으로써 작업장의 최신의 정보를 유지하도록 한다.

또한 최하위의 제어 모듈들은 상태 감시 및 진단하는 모니터링 모듈로부터 기계의 가공 상태, 이상검출, 가공 및 조립품의 흐름을 파악하여 이상이 발생하는 경우, 제어기가 응급 상황에 대처할 적절한 조치를 취하도록 한다.

조영호와 정병호(1999)의 연구에서는 점선 부분, 즉 Dispatching

그림 4. 교육용 CIM구조.

모듈, 조립 제어, 자동 창고 제어 및 부품 반송 제어(컨베이어 벨트 위주) 부분에 대해서 운영 프로그램을 구축하였다. 본 연구에서는 운영으로 표시된 부분, 즉 셀에서의 가공 제어와 원자재, 가공 부품 및 가공이 완료된 부품의 적재/하역, 이송 등을 유기적으로 처리할 수 있도록 RGV, 3-portal, CB에 관한 부품 이송 제어 모듈, 자동 창고 제어를 중심으로 운영 프로그램을 구축하였다.

3. 모델링

자동화된 생산시스템의 동시성과 비동기성을 용이하게 표현하기 위해 Petri-Net이 많이 사용된다(James, 1981; O'Grady, 1987; Zhou, 1995). Petri-Net은 시스템의 충돌 또는 교착 상태 등을 잘 표현할 수 있으므로 동적인 상황을 모델링 하는 데 적합하다(Shinichi and Akihiro, 1996). 이 모델 플랜트를 모델링하는 과정에서 각 개체들 중 특정개체를 공유하는 경우 충돌 현상과 교착 상태 등의 양상들에 대하여 분석 및 조정이 가능해야 한다. 그러므로 본 연구에서는 CB, RGV, 3-portal, MT 간의 충돌, 시스템의 교착 또는 병목 등의 현상을 피하며, 각종 자원의 공유현상을 표현하기 위해 Petri-Net을 모델 플랜트의 모델링 도구로 사용하였다.

제조 셀의 작업물에 관한 Petri-Net 모델에서 정의된 플레이스들은 MT, RGV 등의 가용 여부, 용량 등 각종 자원의 상태를 표현하였다. 트랜지션은 각 플레이스들의 상태 변화를 가져오는 동작이나 사건들로 RGV의 도착과 출발, 3-portal의 적재와 하역 등을 들 수 있다. 가공 지시에 따라 필요한 원자재를 자동 창고로부터 인출하여 I/O\_#2에서 CB\_#4에 투입한다. 투입된 작업물은 CB\_#5가 유희해지기를 기다린다. 작업물이 CB\_#5의 유희 상태를 나타내는 토큰이 발생하면 비로소 CB\_#5에 적재될 수 있다. CB\_#5에 적재된 작업물은 RGV를 기다리고 RGV가 작업물을 적재할 위치(LUP\_#1)에 도착했다는 토큰을 받아야 RGV에 적재된다. 작업물이 MC\_#1에서 가공되는 경우, 작업물을 적재한 RGV는 MC\_#1의 하역위치(LUP\_#2)에 도착하여 3-portal\_#1을 기다리게 된다. RGV로부터 작업물을 적재한 3-portal\_#1은 가공할 MT로 이동한 다음 작업물을 하역하고 유희한 상태가 되어 다음 요구를 대기한다. 작업물이 MC\_#2에서 가공되는 경우, RGV는 MC\_#2의 하역 위치(LUP\_#3)에서 3-portal\_#2가 유희해지기를 기다린다. 유희한 토큰을 받아 점화된 3-portal\_#2는 RGV로부터 작업물을 적재하여 가공 MT에 하역한다. 이때 3-portal\_#2는 유희한 상태의 토큰을 받아 다음 작업을 대기한다. 이러한 트랜지션과 플레이스는 <표 5>에 설명되어 있으며 Petri-Net 모델은 <그림 5>와 같다

표 5. 작업물의 MT 투입 과정에 관한 트랜지션, 플레이스 설명

트랜지션	설 명	플레이스	설 명
T1	작업물 도착	P1	CB #5 가동대기
T2	CB #5 가동 시작	P2	CB #5 가동
T3	CB #5에 작업물 적재 완료	P3	CB #5 유희(idle)
T4	RGV가 LUP #1에 도착	P4	작업물 CB #5 도착, RGV대기
T5	작업물 적재한 RGV 출발	P5	RGV 유희
T6	RGV가 LUP #2위치에 도착	P6	RGV가 LUP #1에서 대기
T7	작업물 적재한 3-portal #1 출발	P7	RGV가 작업물 이동 중
T8	3-portal #1이 LUP #2에 도착	P8	RGV가 LUP #2에 도착, 3-portal #1을 대기
T9	3-portal #1이 MT에 도착	P9	3-portal #1 유희
T10	3-portal #1이 작업물 하역 완료	P10	3-portal #1이 LUP #2 도착, RGV를 대기
T11	MT 가동 시작	P11	3-portal #1이 작업물 이동 중
T12	RGV가 LUP #3에 도착	P12	3-portal #1이 MT에 도착, 하역 중
T13	작업물 적재한 3-portal #2 출발	P13	MT이 작업물 가공
T14	3-portal #2가 LUP #3에 도착	P14	RGV가 LUP #3에 도착, 3-portal #2를 대기
T15	3-portal #2가 MT에 도착	P15	3-portal #2 유희
T16	3-portal #2가 작업물 하역 완료	P16	3-portal #2가 LUP #3 도착, RGV를 대기
T17	MT 가동 시작	P17	3-portal #2 이동 중
		P18	3-portal #2가 MT에 도착, 하역 중
		P19	MT이 작업물 가공

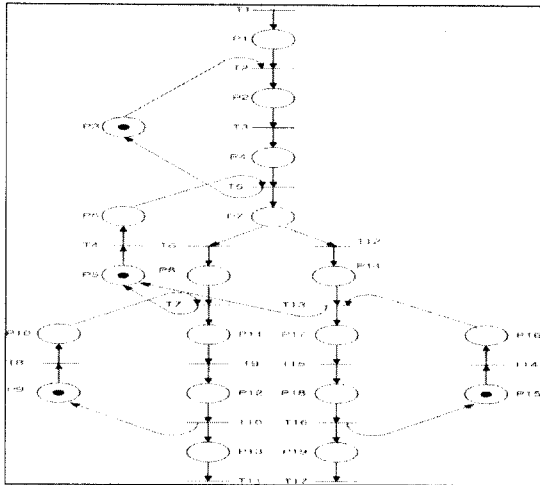


그림 5. 작업물의 MT 투입 과정에 관한 Petri-Net 모델.

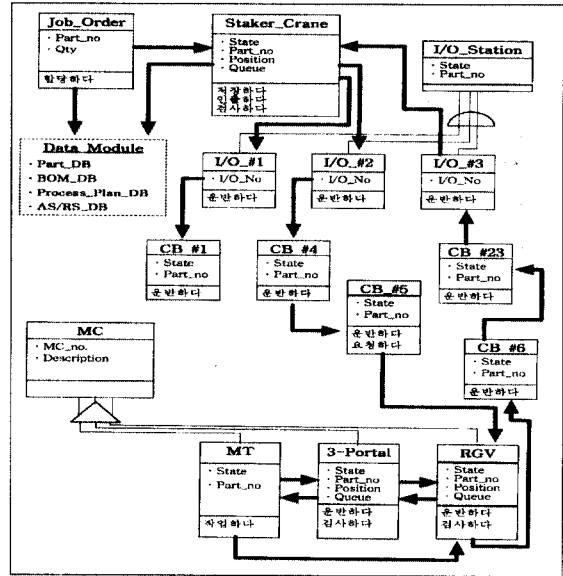


그림 6. 제조 셀의 객체 구조도.

#### 4. 운영 소프트웨어 설계

##### 4.1 객체지향 설계

구의 대상인 모델 플랜트의 CB, 중간부품, RGV, 3-portal, MT, MC 등은 전체시스템을 구성하는 구성품이므로 장비들 간의 통신 및 관리가 무엇보다 중요한 요소로 인식되어야 하기 때문에 장비는 각 개별적인 객체로 취급된다. 그리하여 각 객체의 조정 및 관리는 소프트웨어 개발에 있어 일관성 및 효율성을 유지하기 위한 주요 항목이므로 본 연구에서는 객체지향 기법을 사용하였다. 현재, 객체지향 분석·설계에 많이 사용되는 기법으로는 Rumbaugh(1991) 방법론(OMT, object modelling technique), Booch(1994) 방법론, Coad and Yourdon (1990) 방법론 등이다. 본 연구에서는 제조 셀 운영 소프트웨어를 개발하는데 있어 Coad and Yourdon 방법론을 사용하였는데 이는 Coad and Yourdon 방법론이 시스템의 설계와 개발을 병행하는 다운드 트립형을 채택하고 세세한 제약이 없어 다른 방법론에 비해 범용성이 높기 때문이다.

<그림 6>은 제조 셀을 구성하고 있는 객체 구조도를 보여주고 있다. 스택크레인 클래스는 상태(state), 품번(part no.), 위치(position), 큐(queue)의 속성과 '저장하다', '인출하다', '검사하다'의 메소드를 가진다. Job\_order 클래스로부터 스택크레인 클래스에 작업지시가 할당되면 스택크레인은 Data\_Module 클래스의 정보를 검색하고 해당 품번을 I/O\_#1, I/O\_#2로 인출하며 I/O\_#3에서 완성된 부품이 저장된다. I/O 스테이션 클래스와 CB 클래스는 추상 클래스로 정의되며 각 CB들과 I/O\_#1, 2, 3의 상위 클래스 역할을 하고 Gen-spec (범화-특화, is-a)의 관계를 가진다. MC 클래스는 MT, 3-portal, RGV를 포함하는 Whole-Part(전체-부분, part-of)관계이다.

<그림 7>은 RGV의 '운반하다'에 관한 서비스 차트이다. 서비스 차트는 상태 전이도, 발생 조건, 입·출력 자료, 부작용

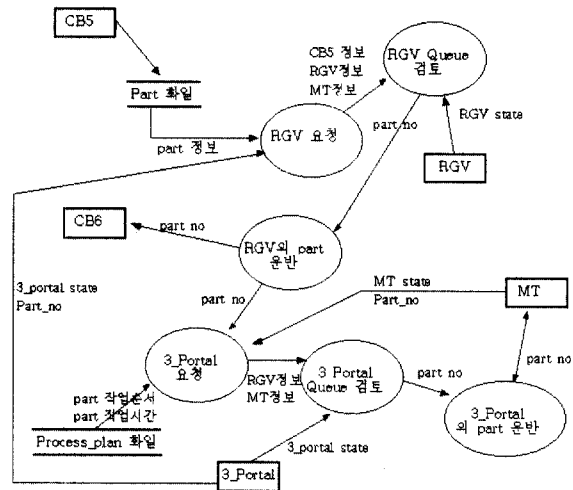


그림 7. 자료 흐름도를 사용해 기술한 서비스 차트-RGV의 '운반하다'.

및 메시지 연결에 관한 정보로부터 서비스 내용을 표현한다. 서비스 내용이란 기본적으로 자료 변환을 의미하므로, 자료 변환에 관한 표현 때문에 전통적인 자료 흐름도가 적합하다 (Shinichi and Akihiro, 1996).

이 서비스 차트는 가공할 부품이 CB\_#5에 도착하면 RGV를 요청하는데, 이때 RGV는 RGV\_Queue로부터 즉시 응답 여부를 검토한다. 만약 선행할 작업이 있다면 RGV\_Queue 증가 먼저 요청한 작업에 대하여 처리한 다음, 해당 순서가 되면 RGV가 작업물을 하역 위치로 운반하게 된다. 또한 MT에서 모든 작업이 끝난 부품은 3-portal을 요청하고 3-portal은 다시 RGV를 요청한다. 그러면 RGV는 다시 RGV\_Queue를 점검하여 해당 순서가 되면 가공이 완료된 작업물을 CB\_#6으로 운반한다.

4.2 클래스 정의

본 논문에서 개발한 소프트웨어에 사용된 대표적인 클래스는 <표 6>과 같다. DB 관련 클래스, 가공 관련 클래스, INTERBUS 제어 클래스, PC-인터페이스 보드에 내장된 펌웨어 4.12를 제어하기 위한 통신 클래스, 전체 프레임을 표현하기 위한 BusConfig 클래스로 나누어진다.

<표 6>은 본 프로그램에서 사용된 대표적인 클래스들을 보여주고 있다. DB 클래스에는 AS/RS DB 클래스, Part DB 클래스, BOM DB 클래스, 가공 데이터 DB 클래스가 있다. 가공부분은 MachCenter 클래스, MachTable 클래스, 3-portal 클래스, RGV 클래스로 구성된다. INTERBUS 제어 관련 클래스에서는 INTERBUS 관련 정보나 실제 작업을 지시하는 클래스들로 데이터베이스로부터 얻은 가공 작업 정보를 가지고 각 센서와 액츄에이터에 어떤 신호를 보내 줄 것인가 결정해 주는 역할을 맡고 있다. 또한 처음 INTERBUS와 신호를 교환하기 위한 노드 핸들(nodehandle)을 얻는 작업과 I/O 모듈의 초기화 작업도 담당한다. 커뮤니케이션 관련 클래스에서는 데이터 처리 작업과 데이터 전송 작업 등으로 실제 디바이스에 보내지게 될 데이터 구조 정의와 펌웨어 정의의 클래스이다. 가공 데이터에 대한 클래스는 IBS RT 24 DIO 16/16-1T 모듈에서 인식할 수 있는 정보들로 구성되어 있다.

표 6. 클래스 분류

분류	클래스명	기능
DB	TASRS	AS/RS Cell 정보와 입·출력
	TPart	품목 기본정보와 입·출력
	TBOM	BOM 정보와 입·출력
	TProcessData	Process Data 정보와 입·출력
Bus Config	TBus	I/O 모듈의 초기화 작업
	TRun	INTERBUS 실행
	TMonitor	Process 상태 모니터링
Communication	TProcessData	Process Data 입·출력
	TData	필드기기 정의와 입·출력
	TlbsController	INTERBUS에서 제공 되는 HLI 관련 메소드
가공 부분	TMachCenter	머시닝 센터에서의 작업
	TMachTable	부품 가공 관련
	T3-portal	MT 및 RGV의 부품 이송과 관련
	TRGV	부품을 MC로 투입 및 가공 완료부품 방출

4.3 데이터베이스 구성

제조 셀을 운영하는 소프트웨어는 자재를 가공하기 위해 상위 레벨로부터 작업 지시를 받아 가공 시스템에 반영할 수 있는 function이 포함되어야 한다. 작업 지시를 받아 가공할 작업물은 해당 자재의 정보와 가공할 장비의 상태 정보 등을 점검하여 자재가 가공될 수 있도록 제어된다. 자동 창고에 보관된 자재에 관한 정보는 자재저장 위치, 형태 등의 재고 정보를 보여주며 가공 또는 조립의 경우, AS/RS DB로부터 쿼리를 이용하여 스테커크레인에 필요한 자료를 제공한다. MT, RGV, CB 등의 장비 정보, 작업 진행 정보, 리소스에 대한 정보는 샵플로어 DB에서 관찰하며 샵플로어 제어기에 관한 제어 정보를 제공한다. 본 논문에 필요한 데이터베이스는 AS/RS DB, Part DB, Part의 작업 순서 및 가공 시간 DB, 각 생산품별 소요되는 부품에 대한 정보가 들어 있는 BOM DB 등이다.

5. 운영 소프트웨어의 개발

5.1 개발 환경

제조 셀의 모델 플랜트 운영을 위한 소프트웨어에서는 자동 창고 셀, 자동 조립 셀, 자동 가공 셀 내의 작업이 서로 독립적으로 수행되어야 한다. 또한 병렬 처리를 위해 각 작업을 멀티 프로세스로 분리하였고, 각 프로세스에서도 독립적으로 작업을 하도록 HLI 클래스를 이용하여 프로그램 모듈을 구성하였다. 사용자 인터페이스 모듈은 객체지향 프로그램인 Delphi 3.0(Borland사)을 프로그램 언어로 사용하여 구축하였다.

INTERBUS와의 인터페이스는 Phoenix Contact사에서 제공한 INTERBUS-s 인터페이스 라이브러리와 펌웨어를 사용하였다. 샵플로어 제어기에서 사용될 각 정보를 저장할 데이터베이스는 DBase IV를 사용하였다. 본 논문에서 개발된 시스템의 환경은 <그림 8>과 같다.

5.2 구축 소프트웨어 소개

본 연구에서 개발된 소프트웨어는 기본 정보 입력, 작업물

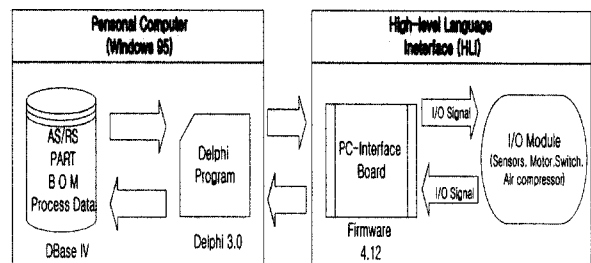


그림 8. 응용 프로그램과 모델 플랜트 간 시스템 환경.

할당 모듈, 수행 모듈의 세 개의 모듈로 구성되어 있다. 기본 정보 모듈은 자동 참고, 데이터베이스 참조, 수정 모듈과 부품 정보 모듈의 입력, 수정 모듈과 BOM 정보 참조 모듈과 프로세스 데이터 정보 모듈로 나뉘어진다. Part의 정보에서는 원자재의 경우 가공경로를 조립품인 경우에는 조립정보를 입력·수정한다. AS/RS 정보에서는 자동참고에 저장중인 원자재, 부분품의 정보가 실시간으로 갱신 처리된다. Sequence 정보 모듈에서는 가공할 Job에 대한 가공순서와 가공시간 정보에 대해 수정 및 참조를 할 수 있다. <그림 9>는 기본 정보의 BOM 정보를 참조하거나 수정할 수 있는 입력 화면이다.

작업물 할당 모듈에서는 기본 정보 모듈에서 입력된 자료를 바탕으로 상위 레벨에서 작업 지시가 내려지면 해당 작업물에 대해 필요한 부품과 소요량을 입력받아 AS/RS 정보로부터 셀의 위치를, BOM 정보로부터 가공할 부품에 대한 원자재와 수량을 산정하고 프로세스 데이터 정보로부터 MT의 가공 순서와 가공 시간을 구한다. <그림 10>은 작업물 할당 모듈에 대한 입력 화면이다. 수행 모듈에서는 모델 플랜트를 구동하기

그림 11. 수행 모듈에 대한 처리 화면.

위한 가공 작업을 진행할 프로세싱 모듈과 모니터링 모듈로 구성되어 있다. <그림 11>에서 프로세싱 모듈에 대한 처리 화면을 보여준다. 프로세싱 모듈은 INTERBUS 컨트롤러를 초기화하고 센서입력부와 모터출력부에 대한 데이터 전송을 준비하여 INTERBUS를 구동하는 모듈이다. 수행모듈은 실제 가공 작업을 맡고 있는 모듈로 작업물 할당 모듈에서 입력된 작업 지시를 바탕으로 먼저 해당 작업물의 자동 참고 셀 위치를 파악하여 해당 품목을 출고해서 가공 라인으로 보내는 자동 참고 로직과 출고된 부품을 해당 MT로 이동 후 가공을 수행하는 가공 모듈 로직으로 되어 있다. 수행모듈에서는 INTERBUS의 내부상황 메시지를 보내주고 있으며 긴급정지, INTERBUS 테스트 등의 기능을 제공한다.

모니터링 모듈에서는 RGV, 3-portal\_Queue, RGV\_Queue 및 처리 상황과 3-portal, CB, RGV, MT등의 장비 상태를 실시간으로 모니터링 할 수 있다.

<그림 12>와 <그림 13>은 모델 플랜트 운영 소프트웨어에 의해 RGV와 3-portal의 운반 작업을 사진으로 보여주고 있다.

그림 12. RGV위의 작업물을 3-portal에 적재.



## 참고문헌

그림 13. 3-portal에 적재된 작업물을 해당 MT에 하역.

<그림 12>는 AGV가 운반해온 작업물을 3-portal이 MT로 운반하기 위해 들어올리는 장면이다. <그림 13>은 가공할 MT에 하역하기 직전의 모습이다.

## 6. 결론

본 논문에서는 INTERBUS를 통해 각 장비들이 연결된 모델 플랜트의 자동 창고(AS/RS) 셀과 제조 셀에 관한 운영 소프트웨어를 HLI 함수를 이용하여 개발하였다. 모델 플랜트에 대한 제조 셀의 동적 상황에 관한 모델링은 Petri-Net 기법을 이용하여 자원의 할당, 용량 문제, workpiece의 흐름 등을 파악하였으며 모델 플랜트 내부에서 발생하기 쉬운 교착 상태, 충돌 현상 등을 고려하여 모델링 하였다. 모델 플랜트의 운영 소프트웨어를 설계함에 있어 Coad & Yourdon 방법론을 채택하여 추후 확장용이하게 하였다.

본 논문은 모델 플랜트를 대상으로 한 교육용 CIM 시스템 개발을 위한 전(前) 단계로, 계속해서 스케줄러를 포함한 플랜트 레벨 제어시스템을 개발할 것이다. 이는 다양한 운영방법에 대한 교육 및 연구의 기반이 될 것으로 본다.

- Booch, G. (1994), Object-Oriented Analysis and Design with Applications, Benjamin/Cummings Publishing Co.
- Coad P. and E. Yourdon (1990), Object Oriented Design with Applications, Benjamin/Cummings Publishing Co.
- Glanzer, David A. and Charles A. Gianfrani (1996), Interoperable Fieldbus devices : a technical overview, *ISA Transactions*, 35, 147-151.
- INTERBUS User manual (1997), Generation 4 Controller Boards as of Firmware 4.12, *Phoenix Contact*
- INTERBUS User Manual (1998), User Interface for High-Level Language N 4 Standard Controller Boards, september, *Phoenix Contact*.
- James L. Peterson (1981), Petri net theory and the modeling of systems, *Prentice-hall INC*.
- Jo, Young-Ho and Byung-Ho Jeong (1999), Development of an Operating Software for a Model Plant usings INTERBUSs, *IE Interfaces*, 12(4), 575-585, December.
- Jones, A., and McLean, C. R. (1986), A Proposed Hierarchical control model for automated manufacturing systems, *Journal of Manufacturing Systems*, 5, 15-25
- Jones, A. and Saleh, A. (1990), A multi-level /multi-layer architecture for intelligent shopfloor control, *International Journal of Computer Integrated Manufacturing*, 3, 60-70
- Liu, Chin-Ming, Chen-Fu Chien and Ho I.-Yung (1998), An object-oriented analysis and for shop floor control systems design method, *International Journal of Computer Integrated Manufacturing*, 11(5), 379-400.
- O'Grady, P. J. (1987), The Control of automated manufacturing systems, *Chapman Hall/ Kogan Page*.
- Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W. (1991), Object- Oriented Modeling and Design, *Englewood Cliffs, NJ: Prentice-Hall*.
- Shinichi Honiden and Akihiro Yamashiro (1996), Development of Object Oriented System, *Dong-il Publishing Co*.
- Smith, J. S. and Joshi, S. B. (1995), A Shop floor controller class for computer integrated manufacturing, *International Journal of computer integrated manufacturing*, 8(5), 327-339.
- Son, S. K., Choi, B. K., Park, T. U. and Lee-Kwang, H. (1991), Resolution of multiple conflicts in the Petri-net model of FAS, *International Journal of Computer Integrated Manufacturing*, 4(4), 253-262.
- Zhou, M. C. (1995), Petri Nets in Flexible and Agile Automation , *Boston: Kluwer Academic Publishers*.

