

자동화 공정 내의 셀 제어기 작동에 대한 정의 및 스케줄러의 개발

도성희*, 이재명**, 이원배***, 박경진****, 강무진*****

Software Development for a Cell Controller Operation and Scheduling in a CIM System

S. H. Do, J. H. Lee, W. B. Lee, K. J. Park, M. J. Kang

Abstract

The demand for automatic manufacturing systems is increasing. One of the crucial obstacles to the Flexible Manufacturing System(FMS) is the lack of excellent strategies for efficient operations. The aim of this research is constructing an automation scheme in the low level of factories where various machineries are involved. An operating strategy is established for an automation unit named as a cell which resides between the upper level computers and manufacturing shop floor. The cell is defined to fit into the total manufacturing system. The defined cell has more functions than conventional cells. A scheduling scheme is adopted for the shop floor operations. A set of software has been developed and tested through simulations and shop floor experiments.

Key Words : FMS, FMC, Cell Controller, Scheduling, Simulations

1. 서론

최근 들어 제품을 생산하기 위한 생산 기술은 여러 사회적, 기술적 문제를 안고 있다. 숙련 기술자의 부

족, 젊은 인력의 제조업 이탈, 제조 설비의 고가화, 제조 업체의 다국적화, 소비자 욕구의 다양화 등으로 제조업 분야는 국가 간의 경쟁이 치열해 지고 있으며, 제조업 관계자는 그 적응을 위한 노력에 필사적이다. 이

* 한양대학교 대학원 기계설계학과
 ** 삼성메이타시스템
 *** 기아자동차주식회사
 **** 한양대학교 공학대학 기계공학과
 ***** 한국과학기술연구원 CAD/CAM 연구실

러한 요구를 충족시키기 위해서 생산 시스템에서의 각 기능 사이의 정보 통합 체계에 대해 여러 모델이 제시되고 있다.” 이들 모델들은 대개 계층 구조로 이루어지며, 그 형태는 Fig. 1과 같이 나타낼 수 있다.

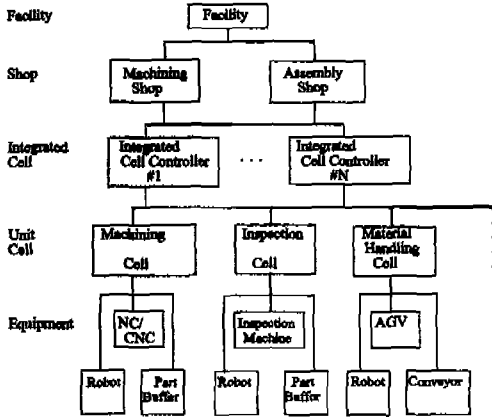


Fig. 1 CIM hierarchy

이러한 생산 시스템을 구축하는 데에는 시스템을 어떻게 구성하고 운용할 것인가 하는 전략의 문제와 이종 기기와 컴퓨터 간의 연결을 담당하는 통신의 문제가 있다. 그 중에서 생산 활동이 직접적으로 이루어지는 셀 레벨과 장비 레벨 즉 하부구조에 대한 연구는 현재 많은 연구가 진행 중에 있으며 이 부분을 셀 제어기 (cell controller, 혹은 워크스테이션 제어기 : workstation controller)라 부르기도 한다. 유연생산시스템 (flexible manufacturing system, FMS)을 추구하는데 있어서 이 셀 제어기는 매우 중요한 역할을 담당하고 있으며 그 접근 방법도 다양하다. 유럽 지역에서는 필요한 기능들을 PC에 분산시킨 뒤 이들을 네트워크 (LAN)로 연결한 형태의 셀 제어기가 있으며,²⁾ 미국에는 계층 구조를 따르기보다는 분산 처리 능력을 향상시킨 셀 제어기라든가,³⁾ 각각의 작업 간의 협조 체제를 강조한 셀 제어기가 연구되고 있기도 하다.⁴⁾

일반적인 개념으로 셀 제어기는 생산 활동에 필요한 스케줄링 등의 계획 기능이나 결정 기능을 가지고 있지 않다. 단지 상부 구조로부터 하달 받은 지시에 의해 생산 활동을 구체화할 뿐이다. 그런데, 소규모 작업장이나 중소기업 등에서는 상부구조의 기능이 축소되어

있고 상부구조의 모든 기능을 필요로 하는 경우는 드물다. 이러한 경우에는 셀 제어기 자체가 자동화의 큰 비중을 차지하게 된다. 그러므로 셀 제어기에 상부 구조의 스케줄링이나 작업 자동실행 등의 일부 기능을 추가하여 셀 제어기 자체를 하나의 축소된 유연생산셀 (flexible manufacturing cell, FMC)로 보아 자동 생산 시스템을 구축하는 것이 바람직할 것이다.

본 연구에서는 셀 제어기 스스로가 유연성을 가지고 생산 활동을 직접 관장할 수 있는 능동적인 셀을 정의하고 정의된 셀의 기능을 실행하는 프로그램을 개발하고자 한다. 즉, 셀 제어기 내로 할당된 작업들에 대하여 장비 레벨에 등록된 기기들의 여건에 맞는 작업 일정 계획을 수립하는 스케줄러 태스크 (scheduler task)와 수립된 일정 계획에 따라 작업들을 자동적으로 실행, 감시하는 작업 자동 실행 태스크 (automatic cell operation task)를 개발하여 모의 실험 및 일련의 실재하는 기기들과의 실험을 통하여 정의된 셀 제어기의 타당성과 실효성을 검증하려고 한다.

2. 셀 제어기의 정의 및 그 운용을 위한 데이터 베이스의 정의

셀 제어기는 앞에서 언급한 바와 같이 장비 레벨에 마련되어 있는 일련의 기기들과 물리적으로 연결되어 이 기기들의 작동을 소프트웨어적으로 관리하는 역할을 담당한다. 즉, Fig. 1의 셀 레벨에 구축되는 일련의 프로그램 군으로서 셀 제어기는 특정 오퍼레이팅 시스템을 보유하는 하나의 컴퓨터 내에 정의될 수 있다. 한편, 셀 제어기는 관장하는 장비 레벨의 성격에 따라 조립 셀, 검사 셀, 가공 셀 등으로 분류할 수 있으나 그 구성은 동일하다고 할 수 있다. 따라서, 본 연구에서는 셀 제어기의 일반적인 구성을 정의하고 타당성을 검토하기 위하여 가공 셀을 대상으로 정의된 셀 제어기의 기능을 실험하고자 한다.

2-1. 셀 제어기의 정의

능동적인 셀 제어기를 위해서는 가장 단순한 형태의 셀 제어기에 대한 연구 및 실험이 필요하다. 즉, 상부 구조로부터 작동에 필요한 정보를 제공 받은 후 이를 그대로 실행하는 역할을 수행하는 셀 제어기가 그것으로서 이에 대한 연구는 몇가지 시행착오를 거쳐 실험하였다.^{5), 6), 7), 8)} 그러나, 정보 전달의 매개체 역할만을 수행

하는 기존의 셀 제어기만으로는 보다 유연성 있는 시스템을 구축하기가 어려울 뿐만 아니라 여러 가지 상황이 벌어지는 현장에 적용하기에는 부적합한 점이 많기 때문에 이를 보완해야 할 필요성이 있다. 따라서 실제 현장 상황에 보다 적합한 셀 제어기를 개발하고자 다음과 같이 정의한다.

우선, 셀 제어기의 유연성을 부여하기 위하여 셀 제어기의 구성을 Fig. 2와 같이 정의한다.^{9,10)} Fig. 2의 셀 영역(cell domain)은 다중 작업 오퍼레이팅 시스템(multi-tasking operating system: 예를 들어 UNIX, VMS, OS/2 등)을 갖는 컴퓨터이며 각 블록은 하나의 독립된 프로그램으로서 오퍼레이팅 시스템 내의 태스크로서 존재한다. 셀 영역의 모든 기능은 일련의 컴퓨터 프로그램들로 구성되는 하나의 소프트웨어 시스템으로 구현되며, 그 프로그램들은 셀의 제반 기능을 관리하는 제어 프로그램(control program)군과 장비 레벨의 기기들과 직접 통신기능을 수행하는 통신 프로그램(communication program)군으로 구분할 수 있다.

모든 태스크는 각각의 고유한 기능을 가지며, 여러 태스크가 서로 상호 작용을 하여 전체적인 셀의 기능을 수행한다. Fig. 2에 정의된 각 태스크의 기능을 간략히 설명하면 다음과 같다.

선 태스크, 스케줄러 태스크, 작업 자동 실행 태스크 등으로 구성되어 있다.

(2) 통신 프로그램 군(communication program group)은 작업 기기들과의 통신을 수행하는 기능을 가지며, 작업 개시 명령 등을 발송 한다든지 기기의 현재 상태 등을 검출하는 역할을 담당한다. Fig. 2에는 표시되지 않았으며 디스패처 태스크들과 프로토콜 변환기 태스크들로 구성되어 있다.

(3) 사용자 인터페이스 태스크(user interface task)는 단말기에 그래픽 도구를 이용하여 사용자가 보다 쉽게 소프트웨어에 접근할 수 있도록 한다.

(4) 컨피규레이션 태스크(configuration task)는 셀 제어기 내부의 모든 태스크들의 동작 상황을 검사하는 기능을 담당한다. 각각의 태스크들의 생성 및 진단 기능등을 수행한다. 본 태스크는 현재 개발되어 있지는 않다.

(5) 스케줄러 태스크(scheduler task)는 셀 영역의 작업 계획(operation scheduling)을 담당한다. 셀내로 할당된 작업들을 실행하기 위한 작업 일정을 수립하는 중요한 역할을 맡는다.

(6) 작업 자동 실행 태스크(automatic cell operation task, ACO)는 수립된 작업일정 계획에 따라 작업을 실행, 감시하는 기능을 갖는다. 이 태스크는 셀 제어기의 운용에 가장 중요한 태스크로서 셀내의 생산 활동을 총괄하여 관리한다. 스케줄러 태스크, 디스패처 태스크 및 프로토콜 변환기 태스크와 작업 정보를 교환하여 작업의 실행 및 감시 기능을 수행한다.

(7) 디스패처 태스크(dispatcher task)는 단순한 프로토콜을 갖는 기기와의 통신을 담당한다.

(8) 프로토콜 변환기 태스크(protocol converter task)는 다소 복잡한 프로토콜을 갖는 기기와의 통신을 담당하는 역할을 수행한다.

(9) 상부 구조 인터페이스 태스크(higher level interface task)는 통합 셀과의 연결을 담당한다. 이 태스크를 통해 상부 구조에서 내려오는 정보를 전달 받는다. 본 태스크는 현재 개발되어 있지는 않으며 여러 셀 제어기를 통합하는 통합 셀 제어기를 구축하려 할 때 개발할 예정이다.

(10) 각각의 태스크들은 데이터 베이스와의 정보를 토대로 운용된다.

(11) 독립적인 여러 태스크와의 상호 정보 교환을 위해서 각각의 태스크와의 통신은 오퍼레이팅 시스템이

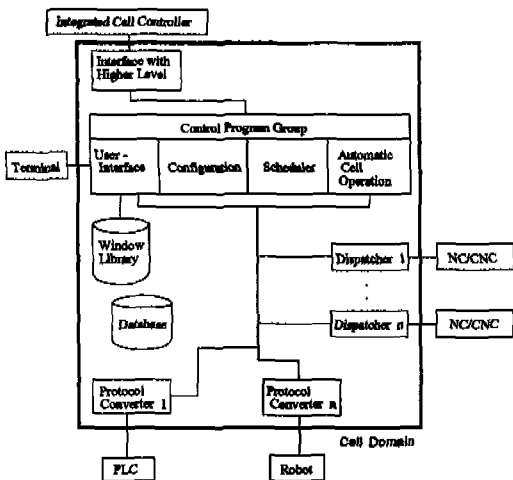


Fig. 2 Cell controller configuration

(1) 컨트롤 프로그램 군(control program group)은 셀 제어기의 운영 및 제어를 담당하며, 컨피규레이션

제공하는 내부 통신(inter process communication, IPC) 방법을 사용한다. 이는 태스크 간의 정보 교환이 특별한 장비(예를 들어 RS232C)를 필요로 하지 않으며 오퍼레이팅 시스템이 동작하는 동안에는 언제든지 그 서비스를 제공 받을 수 있기 때문이다.

중래의 셀 제어기는 단순히 상부 구조에서 하달 받은 생산 정보를 하위의 장비레벨로 하달하여 작업을 실행시키는 피동적인 매개체의 역할만을 하였으나 본 연구에서 정의한 셀 제어기는 스스로가 작업 일정을 수립하고 또한 현재의 작업 진행 상황에 맞추어 계획된 작업을 실행시키는 능동적인 특징을 갖는다. 즉, 상부 구조의 기능 중의 일부를 셀 제어기로 도입하여 상부 구조의 역할이 비교적 축소되어 있는 중소기업 등에서 셀 제어기만으로도 유연 생산을 비교적 손쉽게 실현할 수 있도록 꾀한 것이다.

2-2. 데이터 베이스의 정의

전술한 바와 같이 정의된 셀 제어기는 단순한 정보 전달의 매개체가 아니고 스스로의 결정 기능을 보유하고 있으므로 이를 위해서는 해당 셀 제어기에 적합한 정보를 보관, 관리하여야 한다. 이러한 정보를 바탕으로 셀 제어기 내의 모든 태스크들은 필요한 정보를 입수하고 해석하여 요구되는 기능을 수행한다. 예를 들어 스케줄러 태스크는 스케줄링의 대상이 되는 작업들에 관한 정보와 기기들의 상태에 관한 정보를 입수하여야 하고, 프로토콜 변환기 태스크 역시 해당 기기에 종속적인 독특한 통신 특성에 관한 정보를 입수하여야 통신을 시도할 수 있다. 이와같이, 각 태스크가 필요로 하는 유사한 성격의 정보들을 데이터 베이스화하여 쉽게 정보를 취할 수 있도록 하고 사용자의 필요에 따라 정보의 갱신이 용이하도록 해야 할 필요가 있다.

데이터 베이스를 구축하는 방법은 상용 소프트웨어를 사용하는 등 여러 가지 방법이 있을 수 있으나 본 연구에서는 셀 제어기에서 사용되는 데이터의 특성 등을 감안하여 오퍼레이팅 시스템이 제공하는 디렉토리(directory) 구조를 이용하여 조직적이고 체계적인 데이터 베이스를 정의하였다. 보통의 경우 데이터 베이스를 이용하는 가장 큰 목적은 공동으로 저장되는 공통 정보의 검색, 저장 및 변경 등의 작업을 쉽게 처리하는데 있다. 그러나 본 연구에서 제안하는 셀 제어기의 운용에 필요한 정보들의 처리를 위해서는 앞에서 설명한

데이터 베이스의 기능 외에 각종 기계와의 통신을 위한 실행 화일들을 보관해야 하기 때문에 셀 제어기 자체가 관리하는 데이터 베이스를 정의할 필요가 있다. 한편, 정의된 셀 제어기의 데이터베이스의 상당 부분은 상용 데이터 베이스를 이용할 수도 있으며 이러한 가능성을 배제하지 않고 추후에 상용 데이터 베이스를 사용할 경우에도 쉽게 전이가 가능하도록 데이터 구조를 고려하였다. 따라서 기본적으로 데이터 베이스의 기본 기능은 어느 정도 만족하고 추후에 상용화를 고려한 셀 제어기의 데이터 베이스를 구현한다.

본 연구에서 정의된 셀 제어기에 필요한 데이터 베이스의 구조는 Table 1에 표시하였는데 그 내용을 살펴보면, 제어하고자 하는 정보의 성격에 따라 디렉토리 별로 구분하였으며 그 안에 필요한 정보를 저장하였다. 예를 들어 /Schedule_Data라는 디렉토리에는 스케줄링에 필요한 정보들을 저장하였고 관련된 태스크로는 스케줄러 태스크와 작업 자동 실행 태스크가 있다.

Table 1 Organization of the database

Directory Name	Description	Tasks Concerned
/Configuration_Data	Data for the configuration of a cell	Configuration / User-Interface
/Schedule_Data	Data for scheduling (Data on jobs & operations)	Schedule / Automatic Cell Operation
/Operation_Status	Informations of current status of operations	Schedule / Automatic Cell Operation
/Machine_Data	Informations of machines	Dispatcher / Protocol Converter / Automatic Cell Operation
/Communication_PGM	Communication program with machines	
/NC_Data	NC part programs	
/Robot_Data	Robot programs	
/PLC_Data	Ladder logic programs	
/ICON_Data	Information for the display	User-Interface

3. 스케줄러 태스크에 대한 고찰

유연 생산 시스템에 있어서 생산 관리(production management)는 생산 현장의 능력에 부합하는 제어(capacity control)를 이루기 위한 작업일정계획 정보 구축 즉 스케줄 작업과 이들 작업 일정계획에 따라 작업의 진행 과정을 자동적으로 검출, 보고하는 과정 즉

모니터링 작업이 서로 상호 보완적으로 구축되어야만 바람직한 정보전달 체계가 수립된다고 할 수 있다. 이들 정보 전달의 매개체로는 단순한 셀 제어기가 사용되었으나, 본 연구에서 정의된 셀 제어기는 상부 구조의 역할이 비교적 축소되어 있는 소규모 작업장이나 유연 생산 시스템이 잘 갖추어진 작업장이라 하더라도 상부 구조의 부하를 줄이며 분산처리 방식으로 운영하고자 하는 경우 등에 적용할 수 있으며, 동작에 필요한 기초 정보만 제공된다면 독자적으로 작업을 수행할 수 있도록 하는 기능을 보유하고 있다. 즉, 상부 구조의 기능 중 스케줄 기능을 도입하여 셀 제어기만으로 일정수립 과정을 진행하도록 하는 것이 그것이다.

그러나 기기가 몇 안되는 셀 레벨의 생산 일정을 수립하기 위해 전문적인 스케줄 방법론을 사용하기에는 불합리한 부분이 많고, 조립 셀 및 검사 셀 등과같이 개발된 셀 제어기가 적용되는 작업장마다의 특성을 반영하는 스케줄링을 위해서는 어느 정도 경우에 따른 변경이 가해져야 하기 때문에 본 연구에서 제안하는 스케줄러의 방법론은 기존에 연구된 내용을 참고로 하여 약식으로 작업 일정 계획을 수립하는 방법을 생각한다.¹¹⁾ 다만 작업장의 변화에 쉽게 대응하도록 곳곳에 수정할 수 있는 여지를 남겨 둔다. 한편, 수립된 일정은 계획된 순서에 따라 작업이 진행되도록 하는 작업 자동 실행 태스크의 중요한 입력 자료로 활용된다. 따라서 본 연구에서 제안하는 스케줄러는 셀 제어기 독자적으로 작업을 수행해 나가기 위해 셀 제어기에 적합한 일정을 수립하는 하나의 연결 고리로서의 역할만을 수행하도록 그 기능을 제한한다.

일반적으로 스케줄러 태스크의 주요한 기능은 수행하여야 할 작업들의 구체적인 순서를 정하여 작업을 실행하기 위한 중요한 입력 정보를 제공하는 것이다. 본 연구에서는 스케줄러 태스크를 스케줄링(scheduling) 모듈과 일과표(dispatch list) 생성 모듈로 구분하였다. 스케줄링 모듈에서는 수행되어야 할 모든 작업들의 구체적인 개시 및 종료 시간을 결정하는 작업 일정을 수립하며, 일과표 생성 모듈에서는 수립된 작업 일정과 작업 진행 상황을 기초로 하여 해당 일자별로 개시 혹은 연속 실행되어야 할 작업들을 선별하는 기능을 맡는다.

3-1. 스케줄에 필요한 데이터 베이스

Table 2는 셀 제어기중 본 연구의 개발 대상이 되는 스케줄러 태스크와 작업 자동 실행 태스크의 입력 정보

와 출력 정보를 보관하는 데이터 베이스의 디렉토리 구조를 표현한 것으로 Table 1에서 보여 준 것보다 좀더 정확한 항목들을 나타낸다. 예를 들면 /Schedule_Data/Order 디렉토리는 오더(order)별 공정 계획을 보관하며, Operations 디렉토리는 공정별 데이터를, Dispatch List 디렉토리는 일과표 모듈에서 생성되는 데이터를 보관한다. Work Plan 디렉토리는 스케줄 모듈에서 생성되는 데이터를 그리고 Calendar 디렉토리는 공장력에 필요한 정보를 보관한다.

Table 2. Structure of scheduling and operation status database

Directory Name	Sub Directory	Descriptions
/Schedule_Data	Order	Operations to be executed in the cell
	Operations	Data of operations
	DispatchList	Dispatch lists produced by schedule
	Work Plan	Work plans produced by schedule
	Calendar	Work hours of the dates
/Operation_Status	Status	Informations of operation status
	ExceptionList	Informations of operations which are not coincident with the work plan

한편 이들 데이터 베이스는 상용 데이터 베이스로의 전이 및 공통 정보의 검색, 저장 및 변경 등이 용이하도록 관계형(relational) 구조를 갖는다. 따라서 각각의 디렉토리에 저장되어 있는 정보의 내용을 표현한 것이 Table 4와 Table 5이다. Table 3은 데이터 베이스 태의 각종 항목들에 대한 설명을 알파벳순으로 정리해 놓은 것이다. 이들을 종합해 보면, /Schedule_Data/Order라는 디렉토리 내에 속한 화일(record)들은 오더명(job_id), 선행 오더명(pjb_id), 납기일(dud), 부품명(part_id), 수량(size), 공정갯수(op-no) 그리고 공정명(op_id) 등의 항목(field)에 걸쳐 정보를 보관한다. 또한 /Schedule_Data/Operations라는 디렉토리에 속한 화일들은 화일명으로 오더명(job_id)을 보유하면서 동시에 공정명(op_id), 공정설명(op_name), 기기명(mach_id), 공구명(tool_id), 가공 프로그램명(prg_id), 설치시간(spt) 그리고 순수 가공시간(upt) 등의 항목으로 정보를 저장한다. 한편, 작업이 진행되면서 생성되는 정보들은 /Operation_Status/Status라는 디렉토리 내에 공정명(op_id), 작업상태(status), 기기명(mach_id), 남은 수량(resize), 스케줄 시작시간(sst), 실제 시작시간(pst), 스케줄 종료시간(set), 실제 종료시간(pe) 등의 항목으로 보관된다.

Table 3 Definition of attributes of data

Attribute	Description	Attribute	Description
ARD	Arrival Date/Time	RESIZE	Remained Lot Size
DATE	Date	RSLTT	Result Time
DIFF	Time Difference	SDLT	Scheduled Time
JOB_ID	Job ID	SIZE	Lot Size
LET	Last Event Time	SET	Scheduled End Time
MACH_ID	Machine ID	SPT	Setup Time
OP_ID	Operation ID	SST	Scheduled Start Time
OP_NO	Number of Operations	STATUS	Status of Operation
OP_NAME	Operation Name	TOOL_ID	Tool ID
PART_ID	Part ID	TT	Transporting Time
PET	Practical End Time	TWT	Total Work Time of a Date
PJB_ID	Prior Job ID	UPT	Unit Processing Time
POP_ID	Prior Operation ID	WST	Work Start Time of a date
PRG_ID	Program ID	WET	Work End Time of a date
PST	Practical Start Time		

Table 4 Definition of scheduling database

Directory Name	Contents per Files
Order	JOB_ID, PJB_ID, DUD, PART_ID, SIZE, OP.NO, OP_ID
Operations	OP_ID, OP_NAME, MACH_ID, TOOL_ID, PRG_ID, SPT, UPT
Dispatch List	OP_ID, TT, PRG_ID, MACH_ID, SIZE, POP_ID, PART_ID, STATUS
Work Plan	JOB_ID, PJB_ID, OP_ID, SST, SET, MACH_ID, PRG_ID, ...
Calendar	DATE, TWT, WST, WET

table 5 Definition of operation status database

Directory Name	Contents
Status	OP_ID, STATUS, MACH_ID, RESIZE, SST, PST, SET, PET, LET
Exception List	OP_ID, STATUS, MACH_ID, SDLT, RSLTT, DIFF

3-2 스케줄링 모듈

Fig. 3에서는 제품 생산을 위한 여러 계획 단계(planning stages)들이 설명되고 있는데, 공정 계획(process planning)과 생산 스케줄링(production scheduling)을 포함하는 생산 계획(production planning)은 실제로 제품 생산에 착수하기 전에 이루어지는 모든 계획 과정들을 나타내고, 작업 스케줄링(operation scheduling)과 셀 컨트롤(cell control)을 포함하는 생산 컨트롤은 실제로 제품 생산의 과정에서 이루어지는 작업 내용들이다.

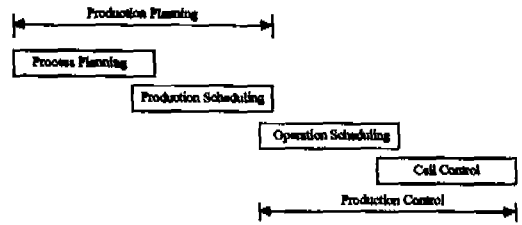


Fig. 3 Planning stages

셀 제어기는 이들 단계중 생산 컨트롤을 담당할 수 있으며 결과적으로 본 연구에서는 작업 스케줄링을 담당하는 스케줄링 모듈을 개발한다. 작업 스케줄링의 목적은 다음의 조건을 만족하도록 작업들의 구체적인 개시 및 종료 시각을 결정하는 것이다.

- 1) 요구되는 납기일 준수
- 2) 총 처리 시간의 최소화
- 3) 기기와 노동 자원 활용의 최대화

작업 스케줄의 대상이 되는 작업장은 여러 가지가 있을 수 있는데 그 예로는 가공을 전문적으로 수행하는 작업장 이라든가. 조립을 전문적으로 수행하는 작업장 혹은 가공과 조립을 병행하는 작업장 등이 있을 수 있다. 셀 제어기에 적합한 스케줄 기능의 검증을 위해서 우선 가공을 전문적으로 수행하는 가공 셀을 대상으로 하는 스케줄 프로그램을 제안하며 가공 셀 스케줄링에 관련된 조건을 다음과 같이 제한한다.

- 1) 하나의 제조오더(job)는 다수 개의 작업(operation)으로 구성되고, 동일한 제조오더에 속해 있는 작업들은 서로 중첩되어 수행될 수 없다. 즉, 선행 작업이 완료된 상태에서만 다음의 작업이 수행될 수 있다.
- 2) 제조오더는 서로 독립적이며 필요한 경우 제조오더 간의 우선 순위는 선행 제조오더명(prior Job ID)을 명시하여 구분한다.
- 3) 주어진 우선 규칙(작업 분배 규칙, priority rule)에 따라 기기가 수행될 작업을 선택한다. 그러므로 데이터 베이스로 부터 입력되는 입력 정보에는 각 작업이 수행될 기기가 명시되어야 한다.

우선 규칙은 스케줄링의 가장 중요한 인자로서, 우선

규칙에 따라 스케줄링의 결과는 상이하게 나올 수 있다. 일반적으로 쓰이는 우선 규칙을 아래에 나열하였는데 본 연구에서는 최소작업시간순(shortest processing time)규칙과 최조잡기순(earliest due date)규칙을 고려하여 스케줄링을 수행한다.

선 착 순 : 기계의 앞에 도착한 것부터
순번으로 가공

최소작업시간순 : 작업 시간이 적은 것부터 가공

최 초 납 기 순 : 납기가 급한 것부터 가공

최소여유시간순 : 여유 시간이 적은 것부터 가공

최대 작업수 순 : 나머지 작업의 수가 많은
것부터 가공

최소 작업수 순 : 나머지 작업의 수가 적은
것부터 가공

최대 페널티 순 : 납기 지연에 의해 생기는 손실
비용이 큰 것부터 가공

Fig. 4에는 본 연구에서 수행한 스케줄링의 개념이 도식적으로 표시되어 있다. 스케줄링 모듈의 입력 자료는 작업의 진행 순서와 재료의 도착 시간(arrival date/hour) 및 납기(due date/hour)가 명시되어 있는 공정 계획(process plan), 스케줄링의 대상이 되는 작업들에 관한 정보(operation data), 사용될 기기들의 상태에 관한 정보(machine data) 및 작업의 우선 순위를 결정하는 우선 규칙(priority rule)등이다. 공정 계획과 작업 정보는 Table 4에 표시하였듯이 /Schedule_Data/Order 디렉토리 및 /Schedule_Data/Operation 디렉토리에서 추출할 수 있으며, 기기들의 정보는 Table 1에서 표시하였듯이 /Machine_Data 디렉토리에서 추출할 수 있다. 또한 우선 규칙은 사용자가 그래픽 화면을 통해 쉽게 입력할 수 있다. 이러한 자료를 바탕으로 시뮬레이션(simulation)을 수행하여 일정 계획(work plan)이 수립되고, 이 일정 계획의 평가 기준이 되는 자료(data for assessment) 즉, 최대 및 평균 체류 시간, 최대 및 평균 납기 지연 값을 계산하여 출력한다.¹²⁾ 여기에서 시뮬레이션은 입력 데이터를 바탕으로 최적의 일정을 추출해 내는 일련의 과정을 의미한다. 예를 들어 특정 작업이 특정기간내에 특정 기계를 사용하고자 할 때 작업의 선행 관계라든지 작업의 우선순위를 고려하여 작업시작일을 계산해 내는 과정을 시뮬레이션이라 한다.

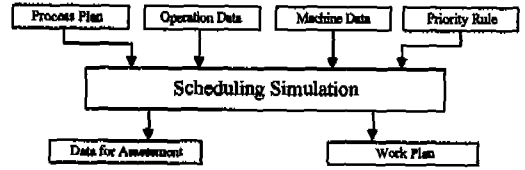


Fig. 4 Data flow for scheduling module

한편, 시뮬레이션시 시간의 변화량은 고정 시간 증분법을 이용하였다. 이는 가변시간 증분법에 비하여 다소 시간이 더 소요될 수 있으나 비교적 코딩이 간단한 이점이 있다. 시뮬레이션시 중요한 것은 시간의 처리이다. 시뮬레이션에는 정수형의 연속적인 시간을 사용한다. 그러므로 재료의 도착 시간과 납기 및 작업의 처리 시간(process time)을 정수형의 시간으로 변환하는 과정이 필요하다. 또한, 시뮬레이션을 통해 결정된 작업의 개시 및 종료 시간을 실제의 일자/시간으로 변환하여야 한다. 이것은 데이터 베이스로부터 즉 Table 2의 /Schedule_Data/Calendar 디렉토리에서 공장력(calendar file)을 입수하여 처리하도록 하였다. 공장력에는 공장의 일자별 총가동 시간, 가동 개시 및 종료 시간이 기록되어 있고 이것을 근거로 하여 정수형의 시간을 실제의 시간으로 혹은 실제의 시간을 정수형의 시간으로 변환할 수 있다. 시뮬레이션의 결과는 Table 4에서 표시한 /Schedule_Data/WorkPlan 디렉토리에 저장되며 추후에 일과표 생성을 위해 사용된다.

3-3. 일과표 생성 모듈

일과표는 해당 일자에 실행되어야 할 작업들의 이름과 그 개시 시간이 기록되는 일종의 목록이다. 일과표 생성 모듈은 스케줄링 작업의 결과로 출력된 일정 계획을 토대로 작업 진행 상황(operation status)과의 비교를 통해 매일 매일의 작업 목록을 만드는 역할을 담당한다.

스케줄링 모듈로부터 생성된 일정 계획만을 토대로 하여 작성된 일과표는 작업이 실행되는 과정에서 기계의 고장등 돌발적인 일이 발생할 수 있으므로 반드시 당초의 일정 계획과 일치한다는 것을 보장할 수 없기 때문에 완전하지 못하다. 따라서, 일정계획만을 의존하여 생성된 일과표는 지연된 작업을 실행시키거나 실행이 중단된 작업을 연속 실행시키기 위한 대책이 없다.

이러한 단점은 작업 진행 상태를 참고하여 현재의 작업 상태를 파악한 뒤 당초의 일정 계획과 비교함으로써 적절한 결정을 내릴 수 있다. 그러므로, 일과표는 작업 진행 상태를 참고하여 당일에 실행되어야 할 작업들의 목록을 작성하는 것이 바람직하다. 이렇게 생성된 일과표는 후에 설명할 작업 자동 실행 태스크의 중요한 입력 정보가 되며 작업 자동 실행 태스크의 결과인 작업 진행 상황 또한 일과표 생성 모듈의 중요한 입력 정보가 된다.

Fig. 5에는 일과표 생성 과정을 도식적으로 표시하였다. 그림에서 볼 수 있듯이 일과표 생성 모듈의 입력 자료는 Table 4에서 설명한 바와 같이 일정 계획(work plan) 정보는 /Schedule_Data/WorkPlan 디렉토리에서, 작업 진행 상황은 /Operation_Status/Status 디렉토리에서 추출할 수 있으며, 그 결과는 /Schedule_Data/DispatchList 디렉토리에 저장되며 추후에 작업 자동 실행 태스크의 입력 자료로서 이용된다.

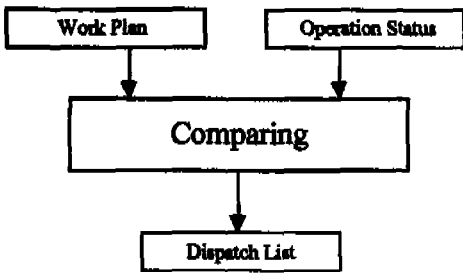


Fig. 5 Data flow for dispatch list module

4. 작업 자동 실행에 관한 고찰

본 장에서는, 수립된 일정 계획에 따라 작업을 실제로 실행시키고 그 상황을 감시하여 작업의 진행 상황을 기록하는 Fig. 2에 표시된 작업 자동 실행 태스크(automatic cell operation task)를 제작하는 방법을 설명한다. 이 프로그램은 수립된 일정 계획에 따라 실제로 작업을 실행시키며 그 진행 과정을 감시하여 기록하는 기능을 수행한다. 작업을 실행시키는 기능은 작업 실행(operation execution) 모듈에서 담당하고 감시, 기록의 기능은 감시(monitors) 모듈에서 담당하도록 하였다.

실제로 작업을 실행시키기 위해서는 기기에 개시 명령을 하달할 수 있는 통신 프로그램이 필요하다. 각종

기기와의 통신은 Fig. 2의 디스패처 태스크나 프로토콜 변환기 태스크를 이용해서 하게되는데 이것은 본 논문의 범위를 넘는 것으로 따로 개발된 통신 프로그램을 이용한다.⁽¹³⁾ 결국, 작업 자동 실행 프로그램은 몇가지의 태스크(작업실행 모듈, 감시 모듈, 통신 태스크 등)이 동시에 서로 정보를 교환하며 자기 기능을 수행하는 구조로 이루어져 있다. 따라서 각각의 태스크 간의 정보 교환을 원활히 지원하기 위해서는 오퍼레이팅 시스템이 제공하는 내부통신기법(inter process communication method)를 사용함으로써 그 문제를 해결할 수 있다. 한편, 태스크 간의 내부 통신을 위해서는 프로그램 개발자가 정의하는 통신 규약이 필요하게 되는데 본 연구에서 정의한 내부 통신 규약은 Table 6에 표시하였다. 각각의 항목을 설명하면 head는 통신 패킷의 시작부를 의미하고 section code는 통신 패킷을 발송하는 태스크를 data field는 보내고자 하는 정보를 의미한다. 정보의 종류로는 대상 기계명(machine id), 부품 번호(part id), 가공 프로그램 번호(program id) 및 각종 정보(information)들이 있을 수 있다. 마지막으로 tail은 통신 패킷의 끝부분을 의미한다. Table 7에는 작업 자동 실행 태스크의 동작에 따른 작업 진행 상태의 기술어(information)들을 나타내었다. 이들의 내용은 작업의 진행 상태에 따라 실행(run), 지연실행(d_run), 종료(end), 지연종료(d_end), 기기정지(m_stop) 그리고 대기 혹은 정지(pause)등으로 구분할 수 있으며 Table 6에 정의한 통신 패킷중 data field에 실려서 각각의 태스크로 전송된다.

작업 자동 실행을 실현하기 위해 각 모듈간의 상호 작용을 Fig. 6과같이 고안하였다. 즉, Fig. 6에 표시한 바와 같이 작업 실행 모듈은 작업 개시 명령을 필요한 정보와 함께 통신 프로그램으로 하달한다. 통신 프로그램은 지정된 기기로 통신을 시도하여 하달 받은 동작을 하도록 하며, 그 결과를 다시 작업 실행 모듈로

Table 6 Definition for inter-process communication packet

Field	Description
Head	STX (0×02)
Section Code	Name of the task which sends message
Data Field	Machine_ID, Part_ID, Program_ID, Lot Size, Information, etc.
Tail	ETX (0×03)

Table 7 Definition of attributes for operation status

Attribute	Descriptions
RUN	Punctual start, Running
D_RUN	Delayed start, Running
END	Punctual end
D_END	Delayed end
M_STOP	Stopping with machine breakdown
PAUSE	Stopping but not completed (Continuous running is needed)

알린다. 작업 실행 모듈은 전달 받은 정보에 따라 필요한 조치를 취한 후 감시 모듈로 정보를 제공한다. 감시 모듈은 전달 받은 정보를 분석하여 기록하고 필요한 경우 작업 실행모듈로 메시지를 송출한다.

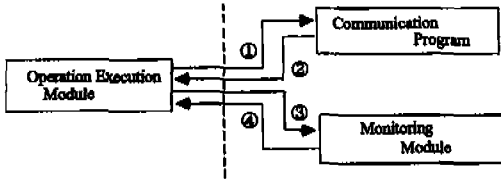


Fig. 6 Message flow of automatic cell operation task

4-1. 작업 실행(operation execution) 모듈

이 모듈은 작업 자동 실행 태스크의 핵심으로서 작업 개시 명령을 통신 태스크에 하달하는 기능과 감시 모듈로 필요한 정보를 전달하는 기능을 갖는다. Table 4의 /Schedule_Data/DispatchList 디렉토리 내의 일과표 파일과 Table 5의 /Operation_Status/Status 디렉토리에 있는 작업 진행 상태(operation status) 그리고, /Machine_Data 디렉토리 내의 기기 정보를 입력 정보로 하여 개시 명령을 내린다. 작업 개시가 가능한 필요 충분 조건은 다음과 같다.

- 1) 예정된 작업 개시 시간이 되었는가?
- 2) 선행 작업(prior operation)은 완료되었는가?
- 3) 사용될 기기는 정상이며, 비어 있는가?

작업 개시 조건을 검사하여 모든 조건이 만족되면, 통신 프로그램에 작업 개시를 명령하고, 이에 대한 응답이 오면 감시 모듈에게 그 결과를 알린다.

4-2. 감시(monitors) 모듈

감시 모듈에서는 작업들의 현재 상태를 검사하여 그

결과를 데이터 베이스에 저장한다. 저장되는 정보는 작업 진행 상황(operation status)과 작업의 이력 상황(exception list)으로서 작업 진행 상황은 수립된 일정 계획과 현재의 작업 진행 상태를 비교하여 계획과 상이한 결과를 보이는 작업들에 대한 내용으로 구성되어 있으며, Table 5에서 표시하였듯이 /Operation_Status/Status 디렉토리에 저장된다. 작업의 이력 상황은 작업자동실행 태스크가 작동되기 시작하면서부터 종료될 때까지의 모든 상황에 대한 내용으로 구성되어 있으며, /Operation_Status/ExceptionList 디렉토리에 저장된다.

이들 결과중 작업 진행 상황은 스케줄러 태스크 내의 일과표 생성 모듈에서 사용되는 정보로서 매일 매일의 작업을 진행해 나가는데 있어서 현장의 상황을 적절히 반영하는데 중요한 정보가 된다. 한편, 이들 내용은 사용자 인터페이스를 통하여 화면에 출력되므로 사용자는 이 두 화면을 참조하여 현재 셀내의 작업 상황을 쉽게 파악하고 셀 제어기가 감당하지 못할 상황이 발생하면 적절한 조치를 취하도록 유도할 수 있다.

5. 소프트웨어의 개발 및 실험

5-1. 개발 환경

본 소프트웨어에서는 오퍼레이팅 시스템 내에 여러 개의 태스크가 동시에 독립적으로 수행되어야 한다. 그러므로 다중 작업(multi-tasking)이 가능한 오퍼레이팅 시스템중 가장 널리 보급되어 있는 유닉스 오퍼레이팅 시스템(UNIX operating system)을 개발 환경으로 하였다. 또한, 작업 자동 실행 프로그램에서 반드시 필요한 태스크 간의 통신 즉, 프로세스(process)간의 통신 수단으로는 UNIX가 제공하는 명명된 파이프 시스템(named pipe system)을 사용하였다.¹⁴⁾

사용자 인터페이스 태스크는 UNIX상에서 사용 기종에 관계없이 모두 통용되는 이식성이 아주 높은 그래픽 도구인 X 윈도우(X-window system)를 이용하여 구축하였다.¹⁵⁾ 프로그래밍 언어로는 C 언어를 사용하였다.

5-2. 프로그램의 운영

본 연구에서 제안된 셀 제어기의 운영은 컴퓨터의 화면상에 나타나는 사용자 인터페이스 윈도우 화면을 사용하여 이루어진다. 따라서 사용자가 Menu를 이용하여 특정 작업을 지정하면 그에 해당하는 작업을 수행하

게 된다. 이러한 일련의 과정을 Fig. 7에 표시하였으며, 수행하고자 하는 목적에 따라 세가지 단계로 구분할 수 있다. 첫번째 단계는 개발된 셀 제어기를 실제 작업장 내의 기기와 물리적인 연결 및 통신 소프트웨어의 설치를 위해 컴피규레이션 태스크를 수행한다. 이 과정은 실제 작업장을 셀 제어기가 인식하도록 하는 과정으로서 아직은 개발되어 있지는 않다.

두번째 단계는 셀 제어기를 정상적으로 사용하는 방법으로 스케줄링을 수행하고 그 결과를 작업 자동 실행이 진행되는 보통의 유연생산셀 기능을 수행한다. 세번째 단계는 스케줄링을 배제하고 직접 셀 제어기내의 각종 통신 태스크들을 이용하여 기기들을 제어하는 방법으로도 사용할 수 있다.

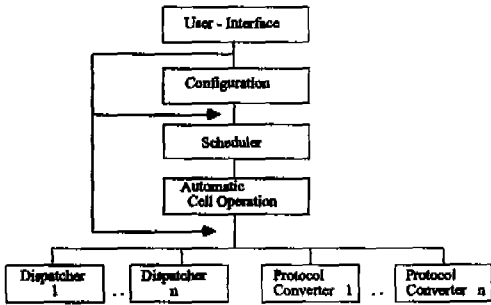


Fig. 7 Management flow of the cell controller

정상적으로 셀 제어기를 사용하는 방법에 대해 설명하면 다음과 같다. 우선, 스케줄링을 수행하기 위해서는 Fig. 8의 화면을 사용하면 된다. 먼저, FileInput 메뉴를 선택하여 스케줄링의 대상이 되는 파일(order file)을 선택한다. 이때 대상 파일내의 정보들의 입력을 위해서 따로 입력을 위한 화면이 필요하나 본 연구에서는 보통의 에디터(editor)를 이용하여 정보 입력을 하기로 한다. 다음으로, ReferenceDate 메뉴를 선택하여 작업의 최초 개시일이 되는 기준일을 입력한다. 스케줄링시에 이 값을 기준으로 하여 필요한 시간 값을 계산하게 된다. 다음으로 Scheduling 메뉴를 선택하여 우선 규칙을 선정하면, 잠시 후에 스케줄링이 완료되어 수립된 일정 계획의 평가기준값(data for assessment)이 그림과 같이 나타난다. 사용자가 적절한 출력 파일명을 그림과 같이 지정하면 스케줄링 과정은 종료한다. 이렇게 하여 저장된 일정 계획의 구체적

인 내용은 FileView 메뉴를 선택하여 볼 수 있다.

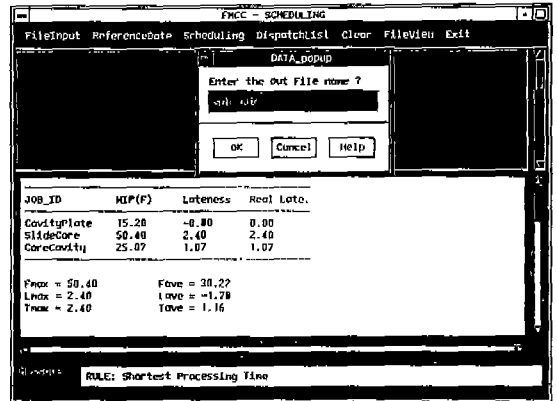


Fig. 8 Result screen of scheduling

일과표를 생성하기 위해서는 Fig. 8의 Dispatch-List 메뉴를 선택한다. 작성하고자하는 일자를 입력하면, 필요한 입력 정보를 얻기 위한 다이얼로그 박스가 차례로 나타나고 적절한 응답을 하면 일과표를 생성하여 그 결과를 Fig. 9와 같이 표시한다. 역시 사용자가 승인하면 일과표 생성 과정은 종료하게 된다. 이러한 일련의 과정에 대한 실제 프로그램은 Fig. 10에 보인 것과 같은 흐름을 갖게 된다. Table 8에 지금까지 설명한 과정들에 사용된 단순한 형태의 예제화일 및 결과들을 나타내었다.

작업 자동 실행은 본 연구의 가장 중요한 부분이다.

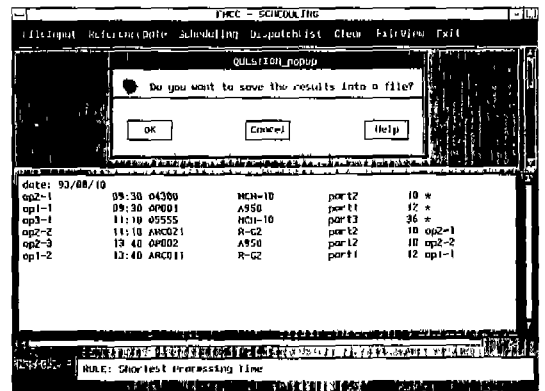


Fig. 9 Result screen of dispatching

작업 자동 실행을 위해서는 Fig. 11에서 보는 바와 같

Table 8 Example data for the scheduling procedure
(a) Order data

File Name	Order_1	Order_2	Order_3
Data Contents	CavityPlate*	SlideCore	CoreCavity
	93/08/10 93/08/13	93/08/08 93/08/14	93/08/09 93/08/12
	part2 10	part3 36	part1 12
	4	3	3
	op2-1 op2-2	op3-1 op3-2	op1-1 op1-2
op2-3 op2-4	op3-3	op1-3	
end	end	end	

(b) Operation data

File Name	CavityPlate	SildeCore	CoreCavity
Data Contents	op2-1 test	op3-1 test	op1-1 test
	MCH-10 04300 tool	MCH-10 0555 tool	A950 OP001 tool
	0m 10m	0m 21m	10m 10m
	/	/	/
	op2-2 test	op3-2 test	op1-2 test
	R-G2 ARC021 tool	R-G2 ARC031 tool	R-G2 ARC011 tool
	0m 15m	0m 21m	0m 21m
	/	/	/
	op2-3 test	op3-3 test	op1-3 test
	A950 OP002 tool	A950 OP003 tool	MCH-10 03100 tool
	0m 15m	0m 11m	0m 14m
	/	end	end
	op2-4 test		
	R-G2 ARC022 tool		
	0m 41m		
	end		

(c) Schedule output

```

JOB_ID : CavityPlate *
op2-1 93/08/10 09:30 93/08/10 11:10 MCH-10 04300 part2 10 *
op2-2 93/08/10 11:10 93/08/10 13:40 R-G2 ARC021 part2 10 op2-1
op2-3 93/08/10 13:40 93/08/10 16:10 A950 OP002 part2 10 op2-2
op2-4 93/08/11 09:52 93/08/11 16:42 R-G2 ARC022 part2 10 op2-3

JOB_ID : SlideCore *
op3-1 93/08/10 11:10 93/08/11 15:46 MCH-10 05555 part3 36 *
op3-2 93/08/11 16:42 93/08/13 13:18 R-G2 ARC031 part3 36 op3-1
op3-3 93/08/13 13:18 93/08/14 11:54 A950 OP003 part3 36 op3-2

JOB_ID : CoreCavity *
op1-1 93/08/10 09:30 93/08/10 11:40 A950 OP001 part1 12 *
op1-2 93/08/10 13:40 93/08/11 09:52 R-G2 ARC011 part1 12 op1-1
op1-3 93/08/11 15:46 93/08/12 10:34 MCH-10 03100 part1 12 op1-2
    
```

(d) Dispatch list

```

date: 93/08/10
op2-1 09:30 04300 MCH-10 part2 10 * WAIT
op1-1 09:30 OP001 A950 part1 12 * WAIT
op3-1 11:10 05555 MCH-10 part3 36 * WAIT
op2-2 11:10 ARC021 R-G2 part2 10 op2-1 WAIT
op2-3 13:40 OP002 A950 part2 10 op2-2 WAIT
op1-2 13:40 ARC011 R-G2 part1 12 op1-1 WAIT
    
```

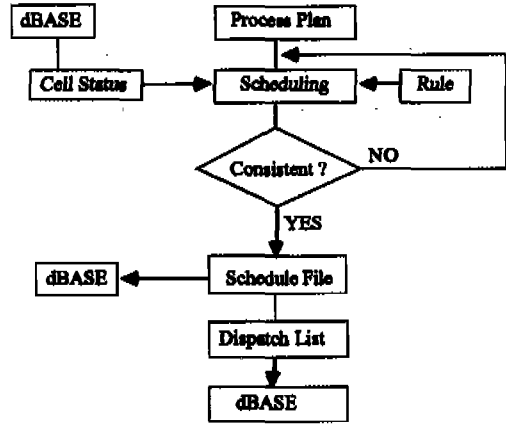


Fig. 10 Flow chart of the scheduler task

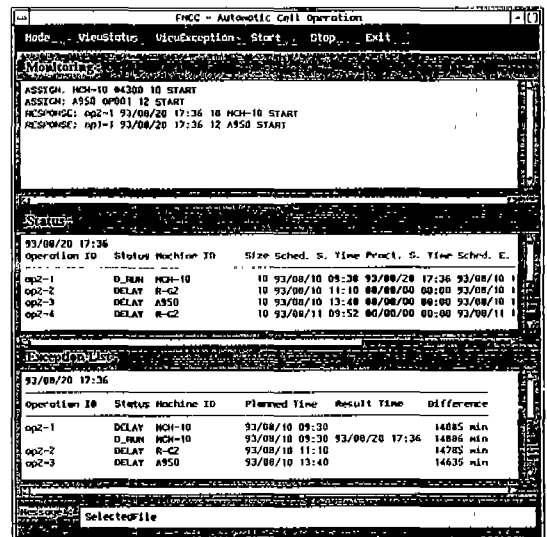


Fig. 11 Screen of automatic cell operation

이 윈도우를 감시(monitring) 영역, 상태(status) 영역 및 이력 상황(exception list) 영역의 세 부분으로 나누어 작업의 진행 추이를 하나의 윈도우 화면에서 쉽게 알아볼 수 있도록 하였다. 이는 몇가지의 태스크가 동시에 작동하기 때문에 이들 태스크 간의 정보 교환을 보다 효율적으로 화면에 출력하고자 한것으로서 작업 실행 모듈과 감시 모듈이 동작하는 상황을 검사할 수 있게된다. 이것은 곧 작업 실행 모듈을 활성화시키

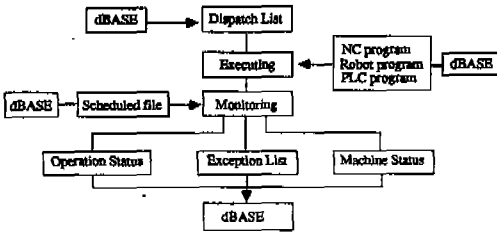


Fig. 12 Flow chart of the automatic cell operation task

고 작업 실행 모듈은 통신 프로그램과 감시 모듈을 활성화시킨다. 이렇게 되면 작업 일정 계획에 따라 작업들이 자동적으로 개시, 실행되며 필요할 때마다 작업 진행 상태 화일(operation status file)과 이력 상황 화일(exception list file)은 자동적으로 갱신이 된다. 작업 실행 모듈이 동작되는 내용은 감시 영역에 자동으로 출력되며, 감시 모듈은 작업 진행 상태 화일이나 상이 내용화일등에 동작되는 내용을 출력하게 되는데 이들의 내용을 보기 위해서는 ViewStatus 와 ViewException 메뉴를 선택하면 된다. 한편, 작업을 중단하기 위해서는 Stop 메뉴를 선택한다. 이때, 작업 실행 모듈은 통신 프로그램으로 작업 중지 명령을 하달하고 중단되었다는 메시지가 올 때까지 기다렸다가 모든 작업이 중단되었음을 사용자에게 알린다. Fig. 12에 는 작업 자동 실행 태스크의 흐름도를 나타내었다.

5-3. 소프트웨어 작동 실험

개발된 프로그램의 작동을 검사하기 위하여 두 가지의 실험 방법이 있을 수 있다. 첫번째 방법으로는 실제의 기기들과 직접 연결되어 있지 않은 상태에서 PC(personal computer)를 가상의 기기로 가정하여 실험하는 방법이며, 두번째 방법으로는 첫번째 방법이 성공한 경우 직접 공장 내의 기기들과 연결하여 실험하는 것이다.

첫번째 방법을 위해서 Fig. 13과 같은 실험 환경을 구축하였다. 즉, PC에 실험대상이 되는 기기와 같은 동작을 하는 에뮬레이터 프로그램(emulator program)을 작성하여 설치하고 셀 제어기 프로그램이 설치된 컴퓨터와 RS-232C를 이용하여 연결하였다.

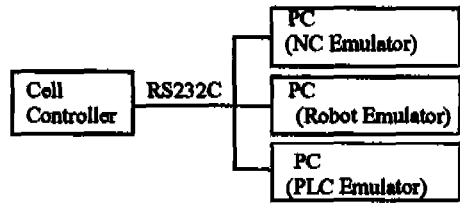


Fig. 13 System construction for simulation

앞에서도 언급하였듯이 통신 프로그램은 기존에 제작된 것을 사용한다. 모의 기기로는 프로토콜 변환기로 기기명 MCH-10인 FANUC Controller를 탑재한 수직형 머시닝 센터와 기기명 R-G2인 FANUC Controller를 탑재한 Arc-mate Robot, 그리고 디스페처 대상 기기로 기기명 A950인 Cincinnati Milacron에서 제작한 5축머시닝 센터등을 대상으로 하였다. 이 모의 실험의 장점은 고가의 기계 대신 PC를 이용하므로 모의 기기의 수를 많게하여 개발된 프로그램의 기능 및 동작을 검증하는 한편 시뮬레이션에도 이용할 수 있다는 것이다. Fig. 8, 9, 11의 화면을 이용하여 실험을 하여본 결과 특별한 상황이 발생되지 않는 한 계획된 스케줄에 맞게 동작이 수행되는 것을 확인할 수 있었다.

모의 실험을 거친 후에는 실제 작업장에서의 실험이 요구되므로 모의 실험의 대상이 되는 기기들과의 직접적인 연결을 시도하였다. 모든 기기가 한 장소에 없는 관계로 우선 기기명 MCH-10과 기기명 R-G2에 대한 실험은 서울대학교 자동화공동연구소에 있는 가공셀 장비들을 대상으로 실험하였고, 기기명 A950에 대한 실험은 한국과학기술연구소(KIST)내에 있는 장비를 대상으로 실험하였다. 이때 작업장에 없는 장비들과의 연결은 앞의 첫번째 방법에 사용되었던 에뮬레이터들을 이용하였다. 이러한 방법을 사용하면 셀 제어기가 보유하고 있는 각종 기능들을 검증하는 데에는 충분하리라고 생각된다. 결과적으로 소규모 작업에 있어서는 적절히 동작하는 것을 확인하였으며 현재 중소기업과 연계하여 현장에 대한 실지 적용 여부에 대한 실험을 계속하고 있다.

6. 결 론

본 연구에서는 자동 생산 시스템인 유연 생산 시스템을 구축하기 위해 전략적인 측면에서 접근하였으며, 셀 제어기의 자동화를 실현하기 위한 방법을 제시하고자 했다. 즉, 종래의 셀이 단지 정보 전달의 매개체 역할만을 수행하던 것에서 탈피하여 상부 구조의 일부 기능을 수용하여 셀 제어기만으로도 소규모 공장의 자동화를 가능하게 하는 방법을 제시하였다. 그 결과 셀 제어기의 자동화에 적합한 셀이 되도록 하기 위하여 셀이 보다 많은 기능을 갖도록 그 구성을 정의하였으며 정의된 셀을 운용하기 위해 필요한 컴퓨터 프로그램을 개발하였다. 아울러 정의된 셀을 효과적으로 운용하기 위한 데이터 베이스를 구축하였다.

셀내로 할당된 작업들의 일정 계획을 수립하는 스케줄 프로그램은 셀에 새롭게 추가된 기능으로서 셀이 능동적인 기능을 갖게 하는 중요한 역할을 담당한다. 상부구조의 상세한 작업 일정에 대한 계획이 없더라도 공정 계획(process plan)만 있다면 셀이 셀내의 기기들의 여건에 부합하는 일정 계획을 수립할 수 있다. 또한 수립된 일정 계획에 따라 작업을 할당, 실행, 감시하는 셀의 작업 자동 실행 프로그램은 셀의 자동 운용을 위한 한 방법이 될 수 있으며, 상부 구조의 지시에 의한 작업 실행이 아닌 셀 제어기 스스로의 결정에 의한 작업들의 실행을 가능하도록 하였다.

참고문헌

1. Rembold, U. and B. Nnaji, "The Role of Manufacturing Models for the Information Technology of the Factory of 1990s", *Journal of Design and Manufacturing*, Vol 1., 1991.
2. Arne Bliberg and Leo Alting, "A Flexible and Integrated Control Concept", *Annals of the CIRP*, Vol 39/1/1990, pp 463-466, 1990.
3. Neil A. Duffie, "Synthesis of the Heterarchical Manufacturing Systems", *Computers in Industry*, Vol. 14, pp 167-174, 1990.

4. Shimon Y. Nof, "Collaborative Coordination Control(CCC) of Distributed Multichine Manufacturing", *Annals of the CIRP*, Vol 41/1/1992, pp 441-445, 1992.
5. Kowalczyk J., Park, G.J. and Rendar R.-B., "Management Scheme of a Workstation Controller in a Flexible Manufacturing System", *Computers in Industry*, Vol 11., pp23-29, 1988.
6. Park, G.J., Lavelle, M.R. and Bruce Rendar R., "Manufacturing Workstations and Cell Controllers in a CIM Application", *Journal of Manufacturing Systems*, Volume 8, Number 2., pp 161-165, 1989.
7. 도성희, 박경진, 강무진, "자동화 공정내의 워크스테이션 제어기 통신 소프트웨어 개발", *한국정밀공학회지*, Vol. 9, No. 1., pp 34-43, 1992.
8. 이원배, 도성희, 박경진, 강무진, "자동화 공정에서 워크스테이션 컨트롤러의 스케줄러 개발", *한국정밀공학회 춘계학술대회*, 1992.
9. 박경진, 도성희, 이재명, 정병수, 강무진, "소프트웨어 개발 관점에서 본 셀 컨트롤러의 설계", *한국정밀공학회 추계학술대회*, 1992.
10. G.J. Park, S.H. Do, S. Her, Y.S. Moon and J. Lim, "Design of a Cell Controller from the Software Development Viewpoint", *Proceeding of the International Fedration of Automatic Control*, Australia, 1993.
11. 人見勝人, 조규갑, "생산시스템공학", *희중당*, 1992.
12. J.H. Mize & J.G. Cox, "Essentials of Simulation", *Prectice-Hall*, 1968.
13. 정병수, 도성희, 박경진, 강무진, "자동화 시스템내 셀 컨트롤러의 통신 소프트웨어 개발 및 실험", *한국정밀공학회 추계학술대회*, 1993.
14. Harvey M. Deitel, "An Introduction to Operating Systems"
15. Oliver Jones, "Introduction to the X Window System", *Prentice-Hall*, 1989.