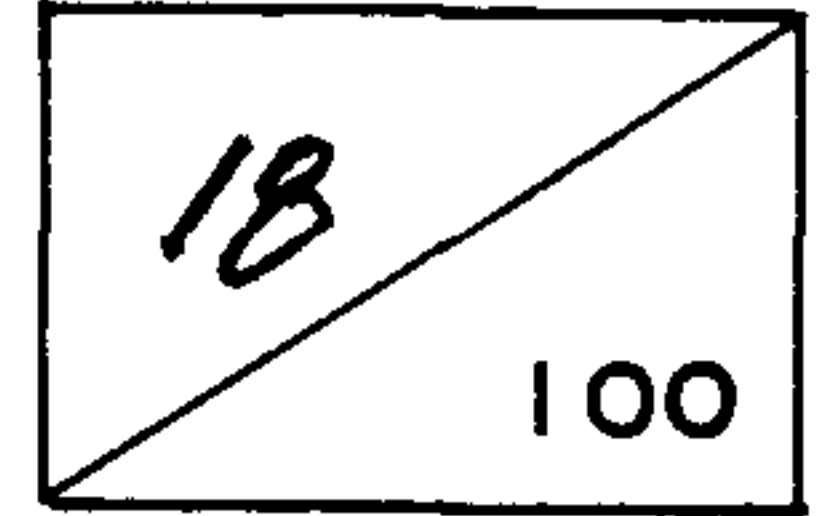


제 3 차 년 도
연 차 보 고 서



컴퓨터를 이용한 통합생산 자동화 기술

Development of CIM Technology

플랜트 통제시스템 개발

Development of a Plant Information System

연구기관
한국과학기술연구원

寄贈	
과학기술처	一九九一年十月三日
寄贈本	

과 학 기 술 처

요 약 문

I. 제 목

플랜트 통제 시스템 개발

II. 연구개발의 목적 및 중요성

생산의 제반 기능인 제품설계, 공정설계, 가공제작 및 생산관리등을 정보처리의 통합화를 통하여 최적으로 조화시키고자 하는 CIM(Computer Integrated Manufacturing)의 관점에서 볼 때 생산통제는 생산관리의 Subset으로써 MRP(Manufacturing Resource Planning)와 함께 생산정보의 흐름을 적시에 적소로 보내주고 제어하는 역할을 한다. 금형공장의 자동화에 있어서 금형이 단일수량 제품이라는 특수성 때문에 각 공정에서의 생산정보가 서로 유기적으로 연계되지 않으면 전체 생산공정의 총화된 효율향상을 성취하기 어렵다. 따라서 이의 해결을 위해 금형 생산의 전 공정을 유기적으로 통합하고자 하는 CIM 기술로의 혁신이 시도되고 있다. 금형 공장내에서는 수 백개의 작업오더가 동시에 처리되는 것이 상례이며 이때 각 제조오더의 납기를 지킴과 동시에 각 기계의 이용율을 최대로 유지하고 생산성 향상 및 원가 절감을 위하여 작업(Order), 자재(Material), 그리고 기계들을 효율적으로 관리·제어 하고자 하는 것이 플랜트 통제 시스템 개발의 목적이다. 이 관리·제어기능은 그 대상, 목적, 계획의 주기 및 정밀도,

Time Horizon 등에 따라 장기, 중기 그리고 단기 기능으로 구분될 수 있다. 금형공장의 운용에 있어 중장기적으로 생산을 계획하고 자재 수급을 관리하는 기능들은 상품화되어 있는 MRP 시스템들이 잘 수행할 수 있는 반면에 작업의 단기적인 분배, 제어 및 최적화 기능은 매우 취약한 형편이며 상품화 되어있는 소프트웨어들도 미진한 형편이다. 금형공장을 모델로한 CIM의 실현을 위하여 Missing Link의 하나인 플랜트 통제 소프트웨어를 개발하여 금형공장 운용의 최적화를 기하고자함이 본 연구과제의 필요성과 목표이다.

III. 연구개발의 내용 및 범위

본 연구는 CIM 기술에 의한 사출금형 생산 모델 플랜트의 통제 시스템을 개발하는 것으로 현재 3차년도 연구 사업을 끝냈다. 먼저 2차년도까지의 연구내용을 정리하면 다음과 같다.

- i) 작업관리 모듈의 Conversion (VAX-Station 이용)
- ii) 생산정보 수집 및 진도관리 모듈의 Conversion
(VAX-Station 이용)
- iii) 작업계획 및 제어 모듈의 Algorithm 개발
- iv) NC 프로그램 관리 모듈 개발
- v) 공구관리 모듈 개발
- vi) 실적관리 모듈의 Algorithm 개발

이상의 작업 내용을 토대로 3차년도에 이루어진 연구내용 및

범위는 다음과 같다.

i) 시스템에서의 기본 Tool 개발

- List Box, Multi-Window 기능등

ii) Database 재구성 및 Database Management Routine 개발

iii) System의 기본 Tool을 사용한 작업관리 모듈의 완성

iv) 작업계획 및 제어 모듈 개발

v) NC 프로그램 관리 모듈의 수정 및 보완 - DNC 기능 개발

IV. 연구개발결과 및 활용에 관한 건의

1. 연구 결과

CIM 기술에 의한 사출금형 공장자동화 사업의 일환으로 구축되는 모델 플랜트의 통제 시스템을 개발 중이다. 1차, 2차년도 연구를 통해 각 모듈의 개발이 완료단계에 이르고 있고 각 모듈 간의 Integration를 통해 상호 유기적인 연결을 꾀하고 있으며, 실제 Process와 생산계획을 잇는 생산 통제 시스템으로 확장, 보완하고 있다.

2. 활용 방안

플랜트 통제 시스템의 각 모듈을 완성해 가는 한편 Machining Center, Robot, 좌표 측정기등의 기기들과의 Communication 문제도 해결하여 금형 생산 모델 플랜트의 On-line Realtime Control System으로 적용한다. 이 시스템은 의사결정자를 위한 지원 시

시스템으로 이용될 것이며 KIST-Industry GMP등을 통하여 회원사의
금형공장에 적용함으로써 β -Test를 거친 뒤 상품화하여 보급될
것이다.

SUMMARY

I. Title of the Project

Development of a Plant Information System.

II. Goals and Significance of the Project

CIM(Compter Integrated Manufacturing) is a new computer application technology to accomplish the optimum solution by combining the manufacturing functions such as product design, process planning, machining, fabrication and process control through the integration of production information. From the point of view of CIM, plant control and MRP(Manufacturing Resource Planning) as a subset of production management distribute and control the manufacturing information adequately. To apply the CIM technology to automation of mold manufacturing, various production orders of molds should be treated to maximize the utilization of machines and productivity and to reduce the production cost through efficient management and control of orders, materials and machines. The management and control function of production can be categorized as long-term, mid-term and short-term functions according to the objects,

purpose, frequency and accuracy of the planning and time horizon. In operating the mold factory, the long-term and mid-term planning and resources management functions can be achieved by applying the commercialized MRP system. On the other hand, the short-term distribution, control and optimization functions rarely can be solved by using commercialized systems. The goal of this project is to develop the short-term plant control software, which is one of the missing links in achieving the automation of mold manufacturing with CIM technology, and to apply the software for optimal management of mold factory.

III. Scope and Contents

The main goal of the project is to develop a plant control system for the CIM model plant of injection mold manufacturing.

And now, the third year of the project has ended. During the previous years, followings are researched and developed.

A. The Second year

- i) Modification and improvement of the order management module
- ii) Development of the algorithms for the job scheduling
- iii) Going development of the NC program management

module

iv) Going development of the tool management module

v) Development of the algorithms for the statistics
module

B. The third year

i) Development of basic tool

ii) Development of the database management routine

iii) Completion of order processing module using by basic
tool

iv) Development of the job scheduling

v) Modification and improvement of the NC program
management module

IV. Results and Recommendations.

The Plant Information System for controlling the model plant as a part of the project "Automation of injection mold factory by CIM technology was developed. And each module of plant information system are ready to integration. The remaining module of the plant control system will be developed in the next year, and the complete system will be applied to a on-line real time control system of the model plant. To do this, communication problems between the plant control system and various machines such as Machining

Center, Industrial Robots, and measuring machines should be solved. The final product will be used as the support system for decision making. Next, this will be commercialized through β -test of GMP(Good Mold Program : industrial consortium of the project member) companies, and distributed.

C O N T E N T S

Chapter 1. Preface	17
1.1. Injection mold production with CIM technology	17
1.2. A position of PIS	22
1.3. System design	25
Chapter 2. Development of basic tool for operaiton in IPIS	28
2.1. Introduction	28
2.2. Function of the basic tool	30
2.2.1. Keyin	30
2.2.2. Pull-Down Menu	33
2.2.3. Scroll Window & Zoom	35
2.2.4. Scroll Text	38
2.2.5. List Box	40
2.2.6. Multi Window	42
2.2.7. Error Check & Message Display	44
2.2.8. On-Line Help	48
Chapter 3. Order management	50

3.1. Introduction	50
3.1.1. The role of the order management ...	50
3.1.2. Structure	51
3.2. Function of the order management	54
3.2.1. Mold	54
3.2.2. Component	60
3.3.3. The course of system integration ...	60
Chapter 4. Job scheduling and Control	63
4.1. Structure of job scheduling and control	63
4.1.1. Management of schedulin in mold factory	63
4.1.2. Structure	67
4.2. Function of job scheduling and congtrol	71
4.2.1. Scheduld Check	71
4.2.2. PERT Calculation	77
4.2.3. Automatic Scheduling	82
4.2.4. Interactive Scheduling	84
4.2.5. Rescheduling	88
4.2.6. Dispatch List Release	101

Chapter 5. Management and distribution of	
the NC program	103
5.1. The role of management and	
distribution of the NC program	103
5.2. Function and Structure of management	
and distribution of the NC program	104
5.2.1. Development environment	104
5.2.2. The role of host computer	107
5.2.3. The role of satellite computer	109
5.3. Development of module in	
host computer	110
5.3.1. DNC configuraiton	110
5.3.2. NC file management	111
5.3.3. NC program transmission	119
5.3.4. Status management	121
5.4. Development of module in	
satellite computer	124
5.4.1. Set the communication protocol	125
5.4.2. NC program teansmit and receive	126
5.4.3. Edit of NC program	127
5.4.4. Interface with host computer	127
5.5. Information link with host computer	
and satellite computer	129

Chapter 6. Development of basic database

management routine	133
6.1. Development of BDMR	133
6.1.1. Introduction	133
6.1.2. Interface with IPIS	134
6.2. Function and Structure of BDMR	134
6.2.1. Structure	134
6.2.2. Function of BDMR	138

목 차

제 1 장 서론	17
제 1 절 CIM에 의한 사출금형 생산	17
제 2 절 플랜트 통제 시스템의 위치	22
제 3 절 시스템 설계	25
제 2 장 시스템 수행을 위한 각종 Tool의 개발 ..	28
제 1 절 Tool개발의 필요성	28
제 2 절 개발된 Tool의 기능	30
1. Keyin	30
2. Pull-Down Menu	33
3. Scroll Window & Zoom	35
4. Scroll Text	38
5. List Box	40
6. Multi Window	42
7. Error Check & Message display	44
8. On-line Help	48
제 3 장 작업관리	50
제 1 절 개요	50
1. 작업관리의 역할	50

2.	작업관리 모듈의 구성	51
제 2 절	작업관리 모듈의 기능	54
1.	금형관리(Mold)	54
2.	금형부품 관리(Component)	60
3.	시스템 통합에 따른 추구 방향	60
제 4 장	작업계획 및 제어	63
제 1 절	작업계획 및 제어 모듈의 구성	63
1.	금형공장의 일정관리	63
2.	모듈의 구성	67
제 2 절	작업계획 및 제어 모듈의 기능	71
1.	Schedule Check	71
2.	PERT Calculation	77
3.	Automatic Scheduling	82
4.	Interactive Scheduling	84
5.	Rescheduling	88
6.	Dispatch List Release	101
제 5 장	NC 프로그램 관리 및 분배	103
제 1 절	NC 프로그램 관리 및 분배의 역할	103
제 2 절	NC 프로그램 관리 및 분배의 기능 및 구성	104
1.	개발 환경	104

2.	중앙 컴퓨터의 기능 구성	107
3.	위성 컴퓨터의 기능 구성	109
제 3 절	중앙 컴퓨터의 모듈 개발	110
1.	DNC Configuration	110
2.	NC 프로그램 관리	111
3.	NC 프로그램 분배	119
4.	현황관리	121
제 4 절	위성 컴퓨터의 모듈 개발	124
1.	Communication Protocol 설정	125
2.	NC 프로그램 전송 및 수신	126
3.	NC 프로그램 편집	127
4.	중앙 컴퓨터와의 접속	127
제 5 절	중앙 컴퓨터와 위성 컴퓨터간의 정보연결	129
제 6 장	Basic Database Management Routine 개발	133
제 1 절	BDMR의 개발	133
1.	BDMR의 필요성	133
2.	IPIS와의 연계방안	134
제 2 절	BDMR의 구성 및 기능	134
1.	BDMR의 구성	134
2.	BDMR의 기능 및 적용에	138

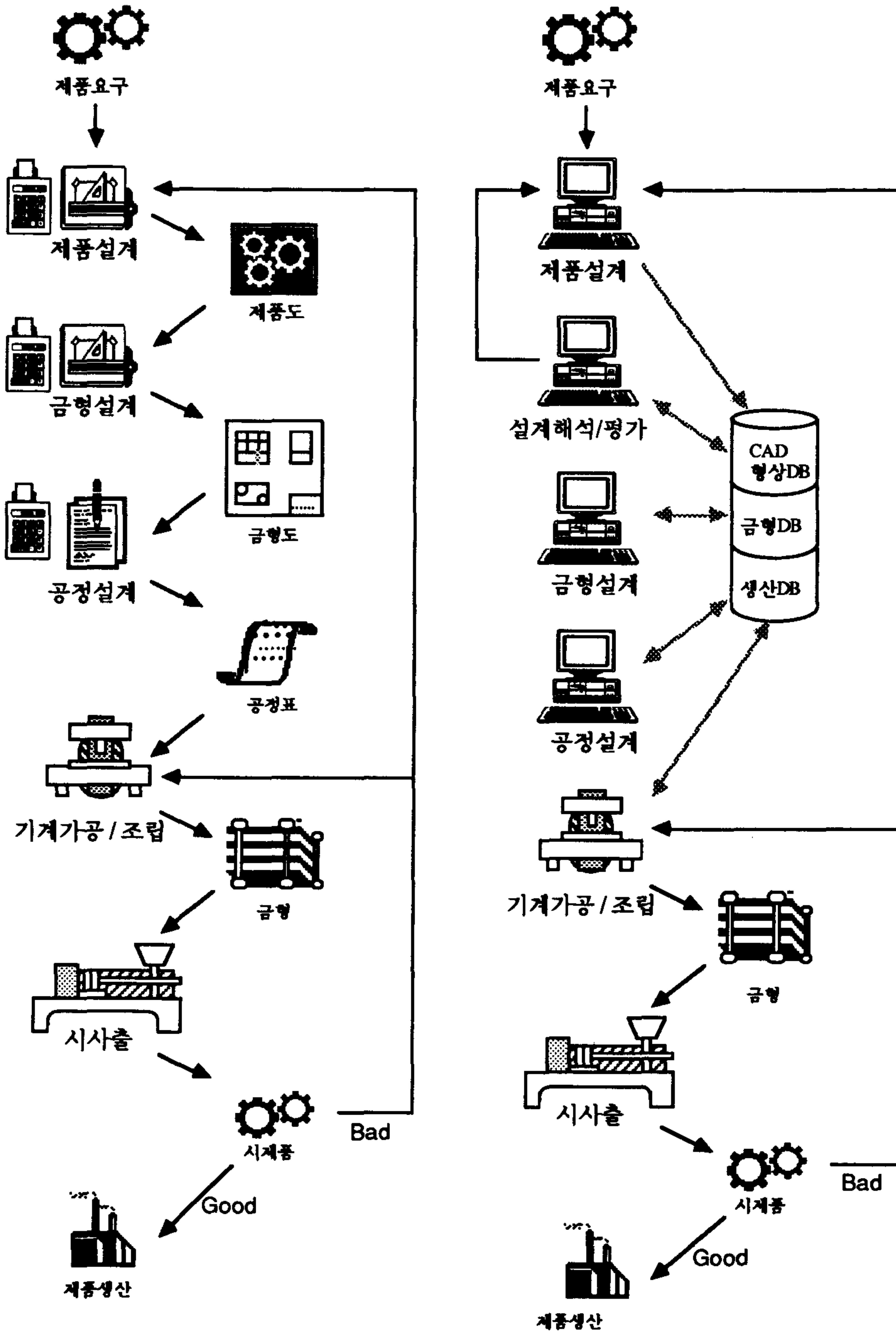
여 백

제 1 장 서 론

제 1 절 CIM에 의한 사출금형 생산

생산자동화가 필요한 분야중에서도, 대량생산을 유발하고 제품생산의 단납기화, 고품질화를 좌우하는 금형기술은 설계에서 가공, 검사 공정에 이르기까지 취급해야 할 생산정보의 양이 방대하고, 실제로 우량의 최종 제품을 얻기 위해서는 많은 기술 축적과 경험을 필요로 한다. 생산정보의 효율적인 처리를 위해 설계와 가공공정에 오래 전부터 컴퓨터가 이용(CAD/CAM)되고 있으나, 금형이 단일수량 제품이라는 특수성 때문에 각 공정에서의 생산정보가 서로 유기적으로 연계되지 않으면 전체 생산공정의 총화된 효율향상을 성취하기 어렵다. 이의 해결을 위해 선진국에서는 금형생산의 전 공정을 유기적으로 통합하고자 하는 CIM기술로의 혁신을 시도하고 있다.

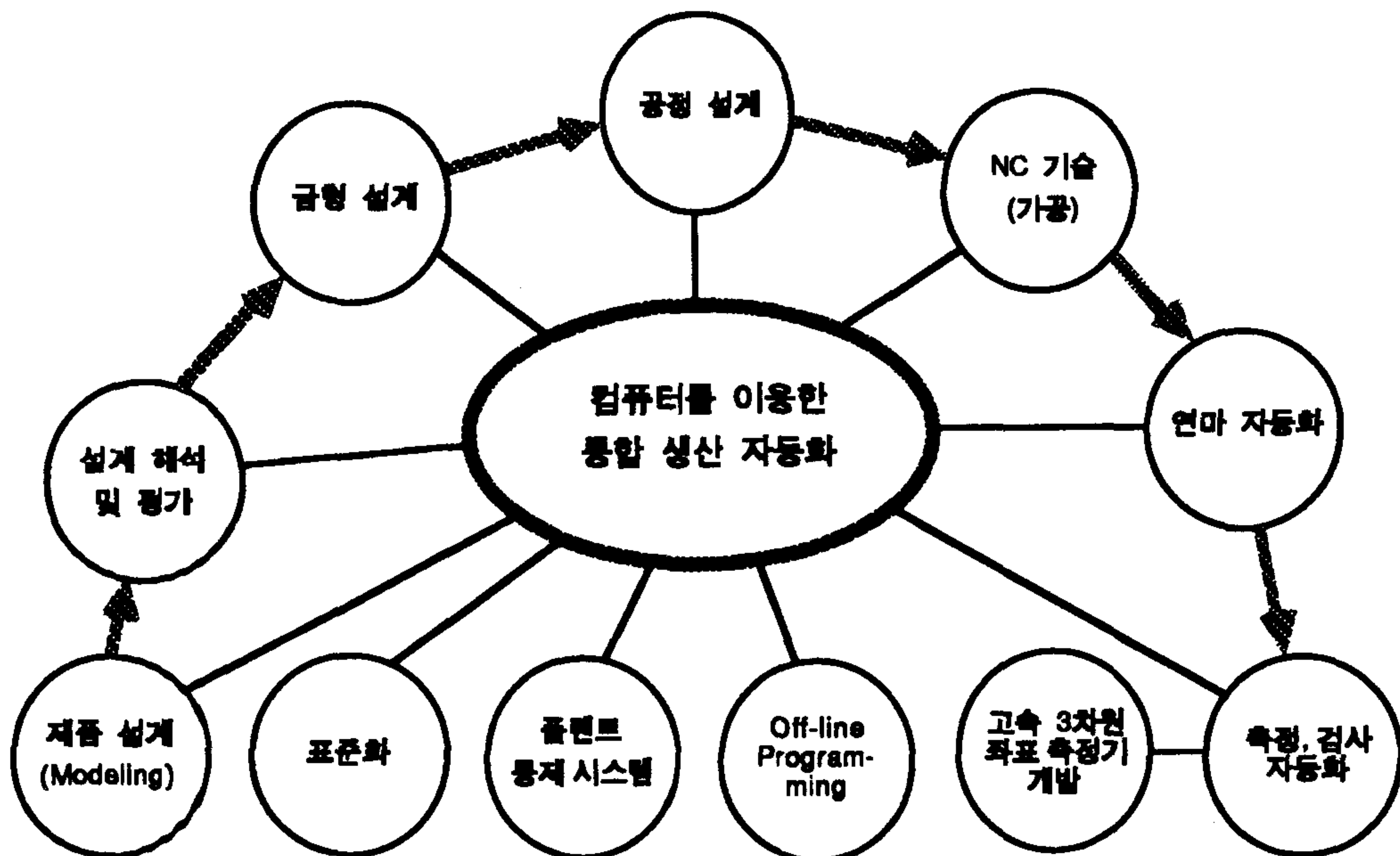
<그림 1-1>은 사출금형에 의한 제품생산의 과정을 보여준다. 종래의 생산방식에서는 각 기능간의 정보교환이 Off-line으로 이루어지며, 금형 제작이 완료되고 시사출을 해본 후에야 설계에 대한 평가가 가능하여 후가공 등의 사후처리적인 방법을 통해서만 그 조정이 일부 가능한 실정이다. CIM에 의한 생산방식에서는 각 기능간의 On-line 연결고리가 형성되어 있어, 잘못된 프로세스가 발생하면 그 영향이 파급되기 전에 Feedback을 통해 사전에



<그림 1-1> 사출 금형에 의한 제품생산의 과정

방하는 식의 생산활동이 전개된다. 예를 들어 제품설계로 부터 금형설계로 넘어가기 전에 설계의 해석·평가 과정을 통해 설계의 적부를 검토한 후 좋은 설계임이 판명되어야 금형이 설계되며, 금형의 가공·연마후의 측정·검사 과정에서 역시 합격판정이 있어야 비로소 조립, 시사출로 넘어가게 되는 것이다. 이와 같은 통합생산에 의할 경우, 발생하는 Error의 확산을 방지할 수 있고 후가공이나 재가공을 최소화할 수 있어 제품요구서에서 부터 제품생산까지의 전체 Leadtime을 획기적으로 단축하여 생산성을 증대시킬 수 있게 된다.

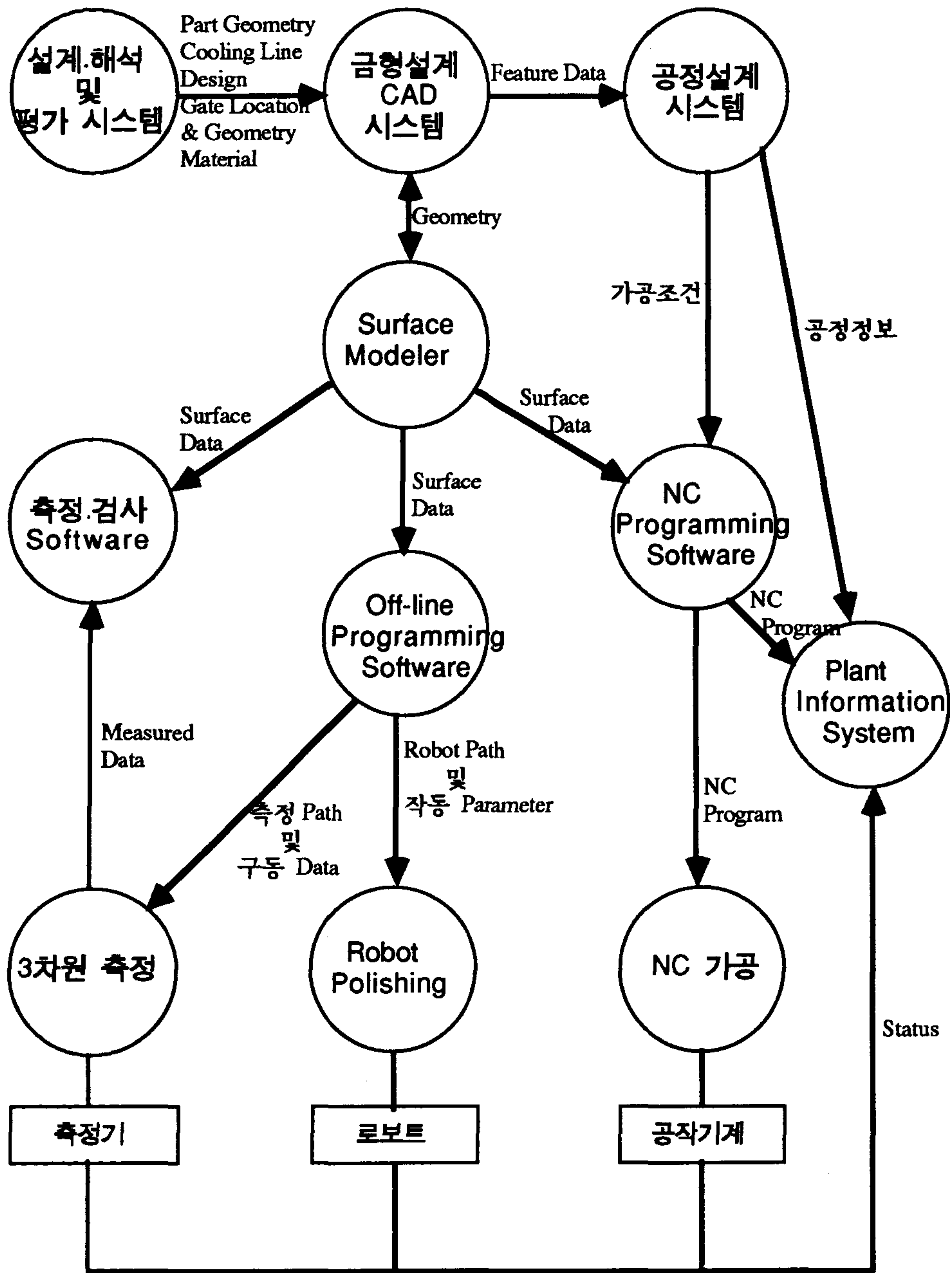
CIM기술에 의한 사출금형 생산자동화의 개념도가 <그림 1-2>



<그림 1-2> CIM에 의한 사출금형 생산의 개념도

에 도시되어 있는데, 반원상에 있는 작은 원들은 좌에서 우로 금형생산의 단계적 과정을 나타낸다. 아래 수평선상에 놓여 있는 표준화와 생산통제, Off-line Programming 등 일련의 기능들은 금형의 통합생산을 보조하고 있다. 사출금형의 생산에 참여하는 이들 기능간의 정보교환이 정보처리 기술에 의해 통합됨으로써 금형 생산의 최적화와 획기적인 납기단축을 이룰 수 있는 것이다.

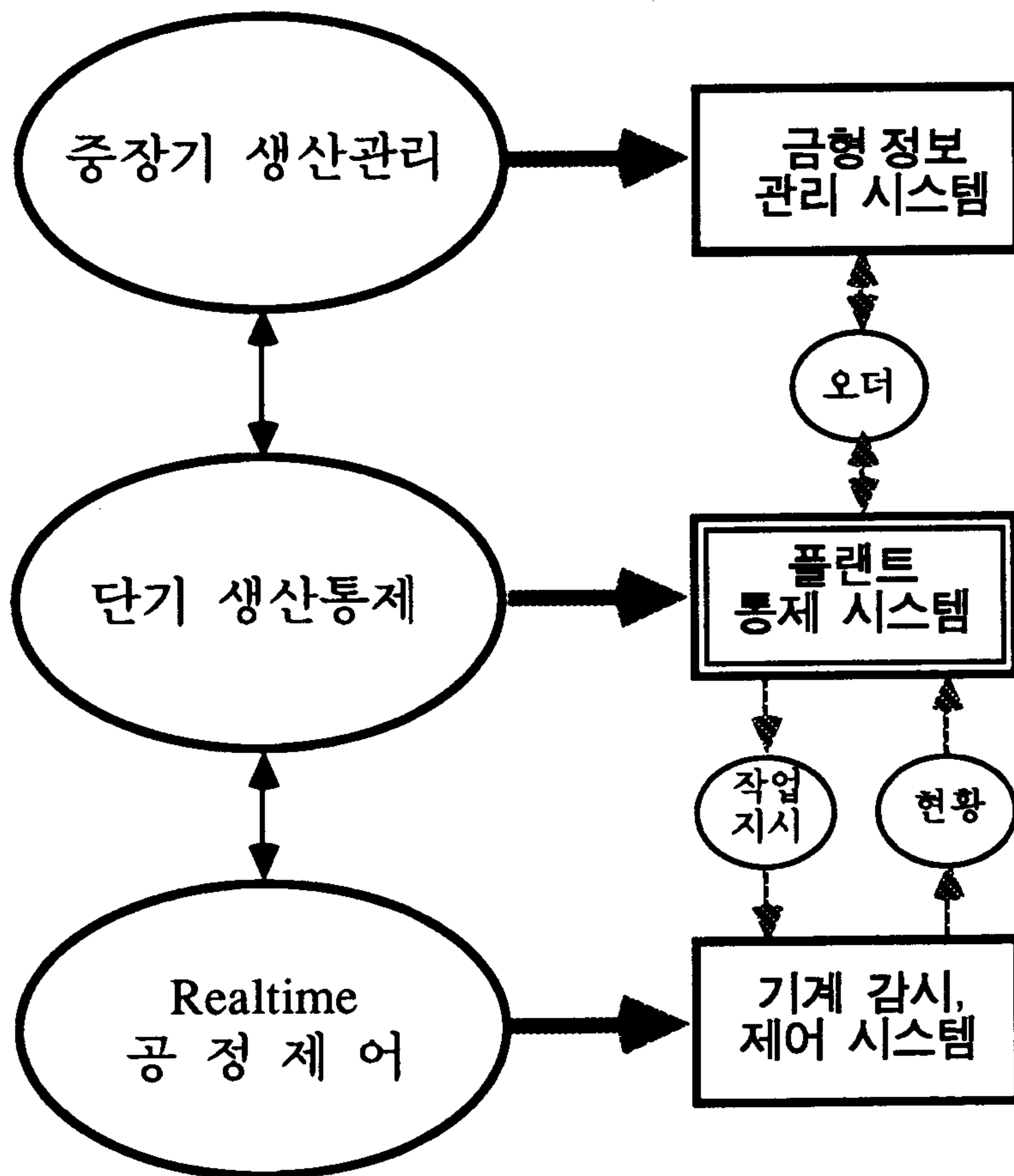
<그림 1-3>은 사출금형 생산에 있어서의 정보의 흐름을 보여 준다. CAD 시스템에 의하여 제품의 설계가 이루어지면 Flow Simulation 및 응력해석 프로그램을 이용하여 유동성, 제품의 강도, 사출후의 품질 등의 관점에서 설계평가가 이루어지고 그 결과가 금형설계 전용 CAD시스템으로 전달된다. 금형설계 CAD 시스템에서는 금형의 표준 데이터베이스등을 이용하여 신속한 금형설계가 이루어질 수 있으며, 금형의 특징(Feature)정보는 공정설계 시스템에서 이용되고, 형상 정보는 그대로 NC프로그래밍 시스템과 연마용 로봇 구동을 위한 Off-line Programming System 및 3차원 측정기의 측정 프로그램 생성 소프트웨어로 넘겨진다. 금형 곡면의 측정 데이터는 CAD시스템의 형상정보와 비교되어 Deviation이 분석되고 다시 가공 또는 연마에 반영되며 공정설계 시스템에서 결정된 가공조건은 형상정보와 함께 NC프로그램 생성에 이용된다. 공정정보는 각 단위기계들의 Status정보 및 품질정보와 함께 플랜트 통제시스템에서 금형생산 플랜트의 최적 운용에 활용된다.



<그림 1-3> CIM에 의한 사출금형 생산에 있어서의 정보의 흐름

제 2 절 플랜트 통제 시스템의 위치

금형공장의 관리측면에서 볼때 플랜트 통제 시스템은 증장기 생산관리를 위한 금형정보관리 시스템과 기계의 작업상태를 감시·제어하는 Monitoring시스템의 사이에 놓여 단기 생산 통제기능을 수행한다. <그림 1-4>에 제시된 바와 같이 상위관리 시스템과의 Interface는 작업오더의 지시 및 보고로 이루어지며, 하위



<그림 1-4> 플랜트 통제시스템의 위치

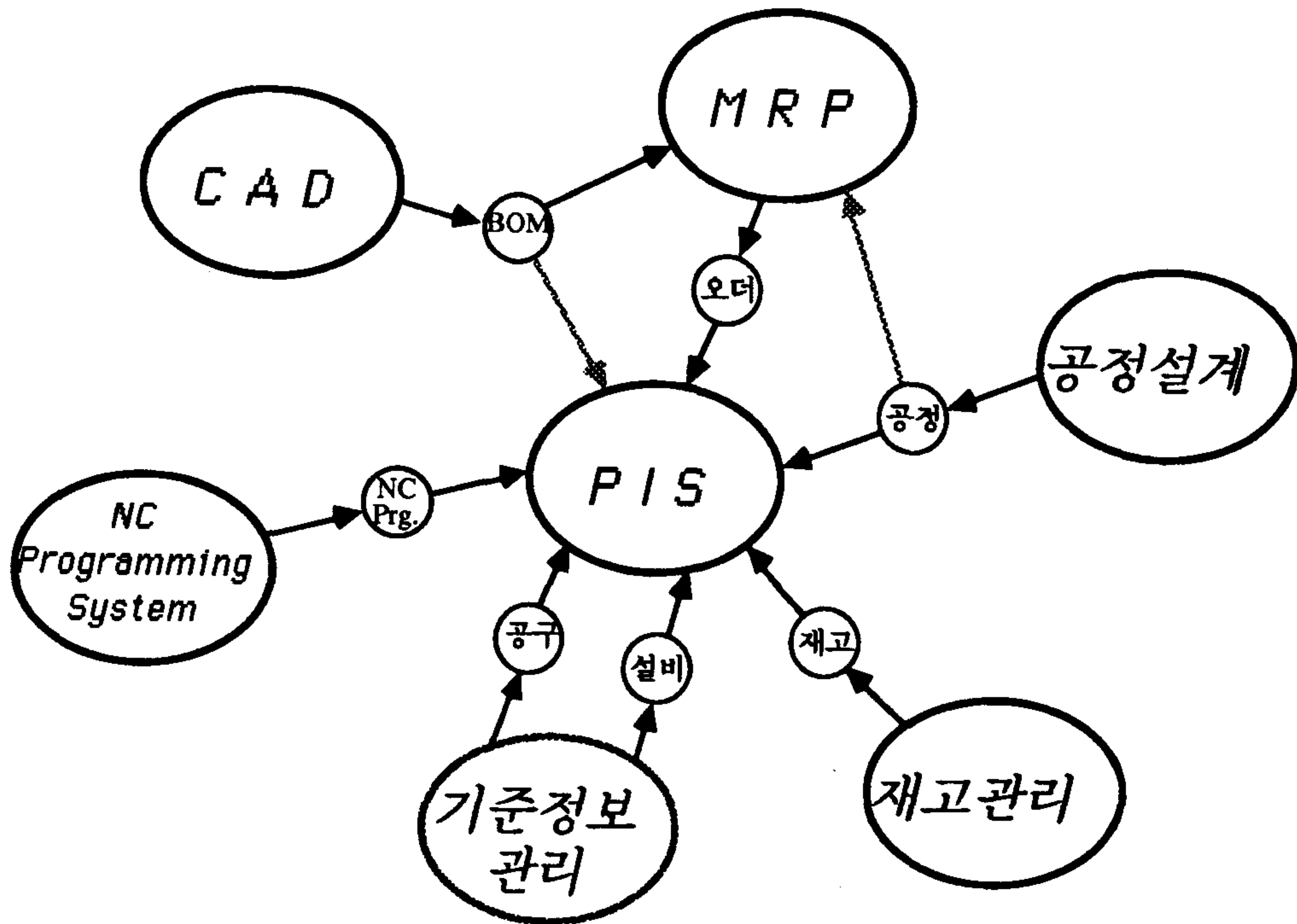
Realtime 공정제어 시스템과는 현황정보를 통한 Communication이 이루어진다. 이 현황정보는

- 작업의 진도 및 실적정보
 - 설비 현황정보
 - 비상사태 관련정보
 - 공구별 사용 실적정보
 - 공작물 - Palette 관련정보
- 등이다.

플랜트 통제 시스템이 처리해야 할 Data중에 생산의 여타 부서에서 제공되어야 하는 것은

- 오더정보
- 공정정보
- 설비정보
- 공구정보
- NC Part Program
- BOM정보

등인데, 이들의 근원은 각각 MRP(Manufacturing Resource Planning), 공정설계, 재고관리, 기준정보 관리, NC Programming System 및 CAD시스템들이다 <그림 1-5>.



<그림 1-5> 플랜트 통제시스템의 Data Interface

제 3 절 시스템 설계

Software Design에 있어서 중요한 문제중의 하나는 주어진 기능을 수행하기 위하여 Software System을 몇개의 모듈로 분할하는 것이 최적인지를 결정하는 것이다. Module수를 적게 하면 각 Module의 크기가 커져 다루기 힘들고 많게 하면 Module의 크기는 작아지나 관리하기가 힘들어지므로 근대의 Software Engineering에서는 소프트웨어 시스템의 Module수와 크기에 있어, 그 개발비

용과 보수·유지비용의 관점에서 어떤 최적치가 존재함을 지적하고 있다. 플랜트 통제 시스템은 처리되는 Data의 Integrity와 기능의 Integrity에 따라

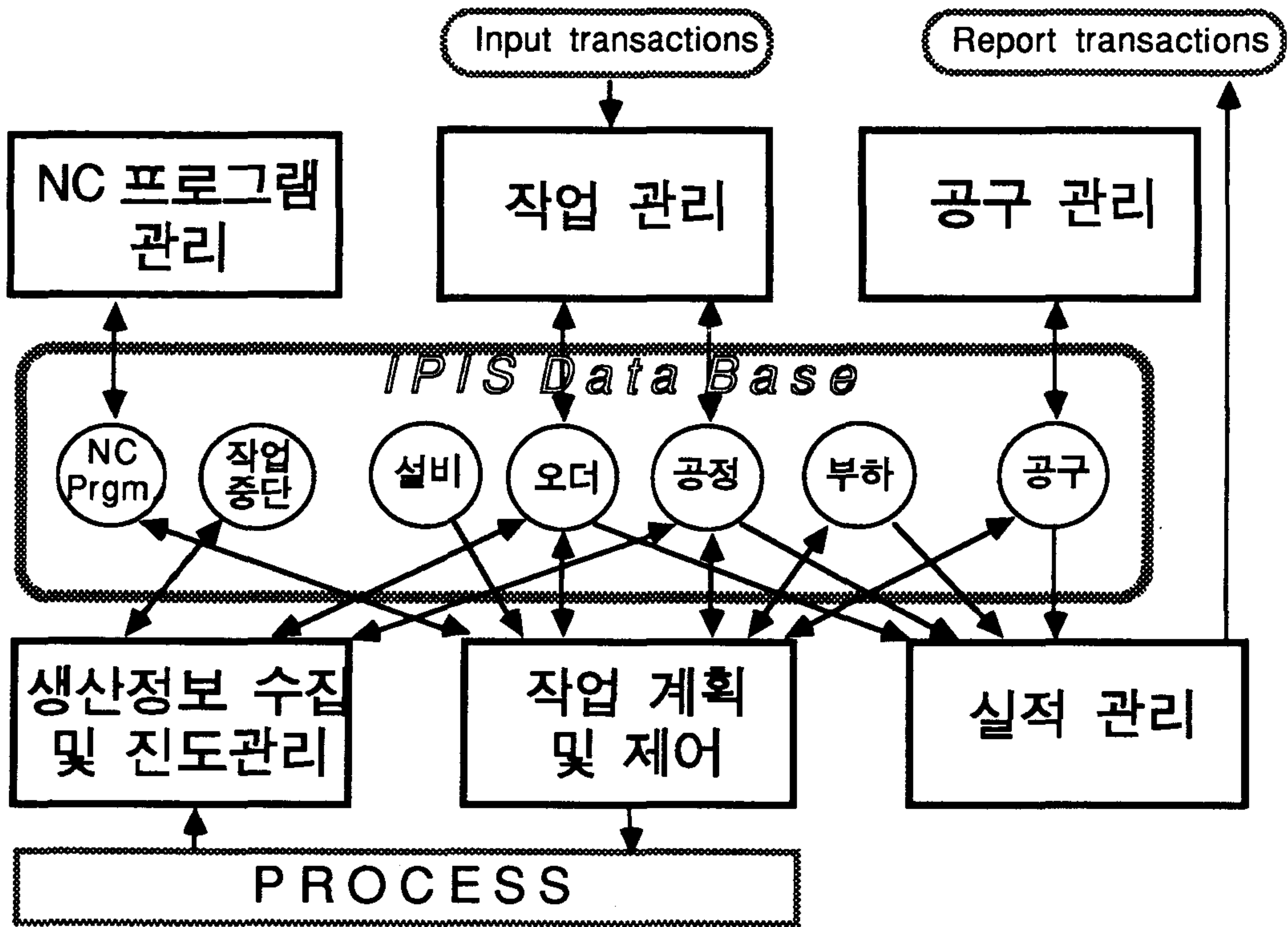
- 작업(Order)관리
- 공구 관리
- NC프로그램 관리 및 분배
- 작업계획 및 제어
- 생산정보수집 및 진도관리
- 시스템 실적관리

의 6개 모듈로 구성하였으며, 각 모듈의 기능은 <표 1-1>과 같다.

<그림 1-6>은 플랜트 통제 시스템내에서의 정보의 흐름 및 Module간의 상호관계를 보여주고 있는데, 각 모듈들이 하나의 PIS Data Base에 Access하고 있음을 잘 나타내고 있다. 작업관리 모듈은 Order정보로써 MRP시스템과 Interface를 형성하며, 작업계획 및 제어 모듈은 Process에 제어정보를 주고, 생산정보 수집 및 진도관리 모듈을 통해 현황정보의 Feedback이 이루어진다.

<표 1-1> 플랜트 통제시스템의 Module별 기능

Module	기 능
작업 관리	작업 Order의 수령, 입/출력, 변경, 삭제 및 작업상태의 보고 NC 프로그램의 수령, 기록, 복사, 소거 및 편집
NC 프로그램 관리 및 분배	NC 프로그램 목록관리, NC 프로그램의 검색, 정지(blocking) 및 준비 (release)
공구 관리	공구수명 감시, 대체공구의 자동관리, 공구 data의 수정 및 관리, 공구 투입을 위한 일정 계획
작업 계획 및 제어	작업 오더의 일정계획 및 부하조정, 작업준비 확인 및 작업지시, 긴급 오더의 추적 및 제어
생산정보 수집 및 진도관리	오더 및 공정의 진행상태/실적정보 및 비상사태 정보집계, 계획과 실제 수행간의 편차집계 및 관리
시스템 실적관리	시스템과 각 구성요소의 통계 data 및 고장원인 집계 및 관리, 보고, 가공비용, Leadtime의 집계 및 관리, 실제부하와 보존작업 data의 보고, 공구수명에 도달한 공구 List



<그림 1-6> 플랜트 통제시스템에서의 Information Flow

제 2 장 시스템 수행을 위한 각종 Tool의 개발

제 1 절 Tool 개발의 필요성

현재 개발되고 있는 대다수의 CIM관련 소프트웨어들을 보면 Graphics Function을 이용하여 사용자가 보다 쉽게 프로그램을 사용할 수 있도록 지원해주는 방향으로 소프트웨어 개발이 추진되고 있다. 즉 그래픽을 이용한 화면구성을 바탕으로 프로그램을 개발하려는것이 주 목적인데, 이러한 수행방법을 Graphics User Interface(GUI)라 부르기도 한다.

GUI를 사용하는 방법은 크게 두가지로 구분된다. 하나는 시스템 영역에 Graphics Library가 준비되어 있어서 프로그래머가 이들 Graphics Library를 자신의 프로그램에서 불러내어 작업 화면을 구성하는 방법이 있고, 또다른 방법으로는 이들 그래픽을 사용하는데 있어서 시스템 영역이 아닌 Utility 형태로 존재하는 몇가지 소프트웨어를 이용하는 방법이 있다. 대다수의 경우에 있어 사용하려는 시스템이 어떻게 구성되어 있는가에 따라 그 방법이 결정되어 지는 것이 보통이며 각각의 방법을 비교해보면 우선 전자의 경우 프로그래머가 일일이 그래픽 요소들을 제어해 가면서 화면을 구성해 나가게 되기 때문에 프로그래머가 원하는 어떠한 형태로든지 화면 구성이 가능한 반면에 최종적인 화면이 나오기까지 많은 시행착오를 거쳐야하는 불편함이 따른다. 후자의 경

우에는 화면을 구성하는 몇가지 Utility를 이용하여 그래픽 요소들을 구성한뒤 이를 프로그램에 이용하기만 하면 되기 때문에 전자보다 훨씬 간편하게 프로그램을 작성할 수 있으나 오히려 사용자가 원하는 특별한 형태의 화면을 구성하기 위해서는 몇가지 어려움이 따르기도 한다.

플랜트 통제시스템을 위한 프로그램은 VAX-Station 이라는 Workstation에서 동작하도록 개발되고 있다. VAX-Station에는 VWS(VMS Workstation Software)라는 Graphics Library가 있으며, 이들 Graphics Library를 사용하여 프로그램을 작성하기 때문에 플랜트 통제시스템에서 사용하는 GUI방식은 앞에서 언급한것중 첫번째 방식을 사용하게 된다. 한편 플랜트 통제시스템의 개발환경은 약 서너명의 프로그래머가 플랜트 통제시스템의 몇가지 모듈을 담당하여 전체 소프트웨어를 개발하고 있는데, 각각의 프로그래머 모두 동일한 방식으로 화면구성을 하고 궁극적으로는 개발된 각각의 프로그램들을 통합하는 작업또한 필요하게 된다. 따라서 프로그램의 개발에 있어서 중복된 투자를 피하고 시스템 통합시 보다 용이하게 통합을 이룩해야하는 등의 문제점들을 해결하고자 다음장에서 설명하는 몇가지 Tool들을 개발하였다.

Tool의 개발방법은 플랜트 통제시스템에서 공통적으로 사용하는 몇가지 기능을 선정하여 이들 각각에 대한 프로그램을 작성한 뒤 프로그래머가 이들 프로그램들을 작성할 필요없이 직접 연결해서 사용하도록 하는 방향으로 하였다. 즉 Graphics Library를 사용하는 시스템에서 그래픽을 구성하는 요소를 Utility화 하여

사용자가 이를 이용하도록 하였으며, 후에 프로그램에 대한 추가 요구가 있을 경우 아주 쉽게 새로운 화면을 구성 할 수 있게 하였다. 프로그램의 아이디어는 Utility Program들을 이용하는 시스템에서 사용되는 화면구성을 참조로 하였다. 즉 Macintosh에서 동작하는 여러 프로그램등의 실행화면에서 사용되는 Window나 Dialog Box 혹은 DOS에서 동작하는 MS-WINDOW등에서 사용되는 그래픽 화면들을 참조로하여 각각의 그래픽 요소들을 구성하였고, 그 기능면에서도 기본적으로는 같은 기능을 가지는 방향으로 추진하였다. 이렇게 함으로서 시스템 통합의 측면이나 프로그램 개발시 화면구성에 대한 중복투자의 문제점들은 어느정도 해결되리라 기대된다.

제 2 절 개발된 Tool의 기능

다음에 열거하는 Tool들은 VMS를 오퍼레이팅 시스템으로하고 VWS라는 Graphics Library를 탑재한 시스템에서 동작되는 프로그램들이다. 프로그램은 'C' 언어를 이용하여 작성하였으며, 프로그래머가 원할경우 Link 작업 만으로도 원하는 기능을 수행할 것이다.

1. Keyin

어떠한 시스템일지라도 기본적으로 데이터를 입력시키기 위해

서는 Keyboard를 사용한다. 플랜트 통제시스템에서도 각각의 모듈에서 데이터를 입력받기 위해서는 Keyboard 입력기능이 필요하다. 그러나 VWS에서 지원하는 그래픽 윈도우들은 IBM-PC계열의 컴퓨터에서처럼 자유자재로 Graphic 모드 혹은 Text 모드로의 전환이라든지 Graphic 모드에서의 Text 수행기능과 같은 방식을 지원하지 않기 때문에 이를 위한 Tool의 개발이 필요하다. 따라서 그래픽 윈도우상에서 사용자가 원하는 정도의 Keyboard 작업을 지원하기 위해서 Keyin이라는 Tool을 개발하였다.

Keyin에서 지원하는 기능으로는 다음과 같은 것이 있다. 우선 보통 데이터 입력에 사용되는 Alphanumeric Character를 받아들여 Integer, Real, String, Character등으로 변환한다. 한편 Keyin에서 지원하는 기능 Key로는 VAX 시스템의 Command에서 삽입 혹은 삭제를 위한 Toggle기능을 갖는 Ctrl A Key, ←→↑↓ 등의 Arrow Key, Return, Automatic Skip, Del Key등이 있다.

Keyin을 사용하는 보조기능 즉 프로그래머가 입력시키는 각종 정보로는 Text의 색상지정을 위한 Foreground/Background Color, Text의 형태지정을 위한 Font, Text가 위치하는 Position, String Size, 대소문자 등과 같은 Character Case, 초기 입력을 위한 Default Input등이 있다.

Keyin이 사용되는 용도는 하나의 데이터 입력 혹은 몇가지 관련된 데이터 군들의 각각의 Field에 대한 입력에 사용된다. 프로그래머는 Keyboard 입력에 대한 화면구성을 통해 각각의 Field에 대한 조건(예를들어 Size, 대소문자, Color, ...)등을 정하여 프

로그인을 개발하며 사용자는 적절한 시기에 Keyboard 입력을 행하면 된다. 한편 기본적으로 데이터 입력시에는 VAX에서 사용하는 삽입 혹은 삭제등과 같은 모든 Keyboard 동작이 가능하며 좌우 Arrow Key를 사용하여 하나의 Field내를 움직일 수 있다. 데이터 군들의 입력시에는 각각의 Field를 상하 Arrow Key를 이용하여 이동이 가능하며 사용자가 원하는 Field로 이동하여 수정 작업을 행하게 된다.

Keyin은 빈번한 Keyboard 작업에 상당히 유용한 Tool로서의 역할을 충분히 수행하리라 기대된다. Keyin의 예는 <그림 2-1>에 표시하였다.

INITIAL DATA INPUT OF MOLD RELATION SYSTEM
OPER. PROTECTING

Initial Data Input of Mold Relation

Mold ID Number : SS 101
Mold Description : TELEPHONE MOLD WIRELESS FORM

Customer Company Number : SAM SUNG
Mold Start Date : 91/09/01
Mold End Date : 91/12/31
Material Receipt Date : 91/09/10
Mold Drawing Receipt Date : 91/09/11
Planned Start Date : 91/09/15
Planned End Date : 91/12/10
Total Cost of the Mold : 100000
Mold Priority : 1

1991
SEP
3
TUE

<그림 2-1> Keyin을 이용한 Data 입력

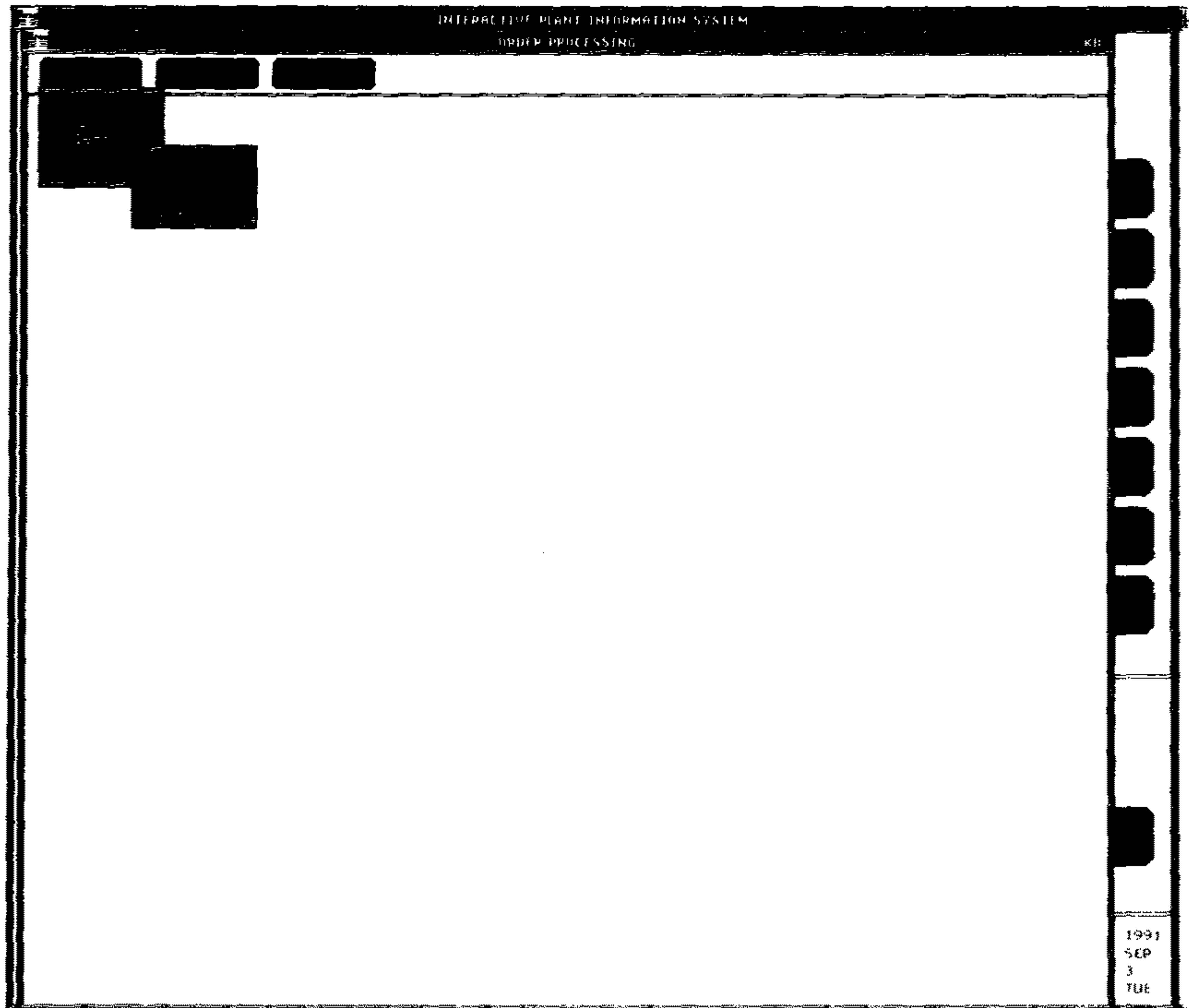
2. Pull-Down Menu

그래픽 화면을 이용한 소프트웨어에는 여러종류의 Menu 시스템이 요구된다. Pop-Up Menu라든지 Pull-Down Menu라 불리는 것이 그 예이며 각각의 Menu마다 장단점이 존재한다. Menu 시스템을 도입하는 방법에 있어서는 개발하려는 소프트웨어의 특성이 가장 많이 좌우하게 되는데 소프트웨어의 성공유무는 적절한 Menu 시스템의 적용에 좌우된다고 해도 과언이 아닐것이다.

플랜트 통제시스템에서는 기본적으로 Pull-Down Menu 시스템을 사용한다. 다만 초기화면에서는 각각의 모듈을 선택하는 부분이 일반 Menu 시스템과 다른데 각각의 모듈마다 방대한 작업을 행하기 때문에 플랜트 통제시스템에서 추구하는 방향을 나타내고자 따로 분리하였다. Pull-Down Menu 또한 각각의 모듈에서 중점적으로 사용하는 Tool로서의 역할이라 할수 있으므로 개발의 대상이 되었다.

개발된 Pull-Down Menu는 VAX Workstation에서 사용하는 Menu 시스템과 동일한 방법을 사용하는 방향으로 개발하였다. Pull-Down Menu가 가지고 있는 기본적인 기능으로는 Mouse Pointer의 이동에 따른 색상의 반전과 Mouse Button을 이용한 Select가 있다. 한편 하나의 Pull-Down Menu에서 또다른 Sub Menu를 계속적으로 생성시킬수 있으며 Menu위에 Working Window의 Overlap 발생을 방지하는등의 기능또한 포함하고 있다. Pull-Down Menu를 사용하기위하여 프로그래머는 Working Window

의 ID Number, Menu String, 메뉴의 갯수, 메뉴가 위치할 좌표등을 제공하며, 입력된 정보에 따라서 자동적으로 Pull-Down Menu가 생성된다. 사용자는 Mouse를 사용하여 원하는 작업을 실행하는데, Pull-Down Menu에서는 Mouse의 움직임에 충실히 반응함과 아울러 사용자의 선택에 적절한 대응을 하도록 설계 하였다. Pull-Down Menu의 예는 <그림 2-2>에 나타내었다.



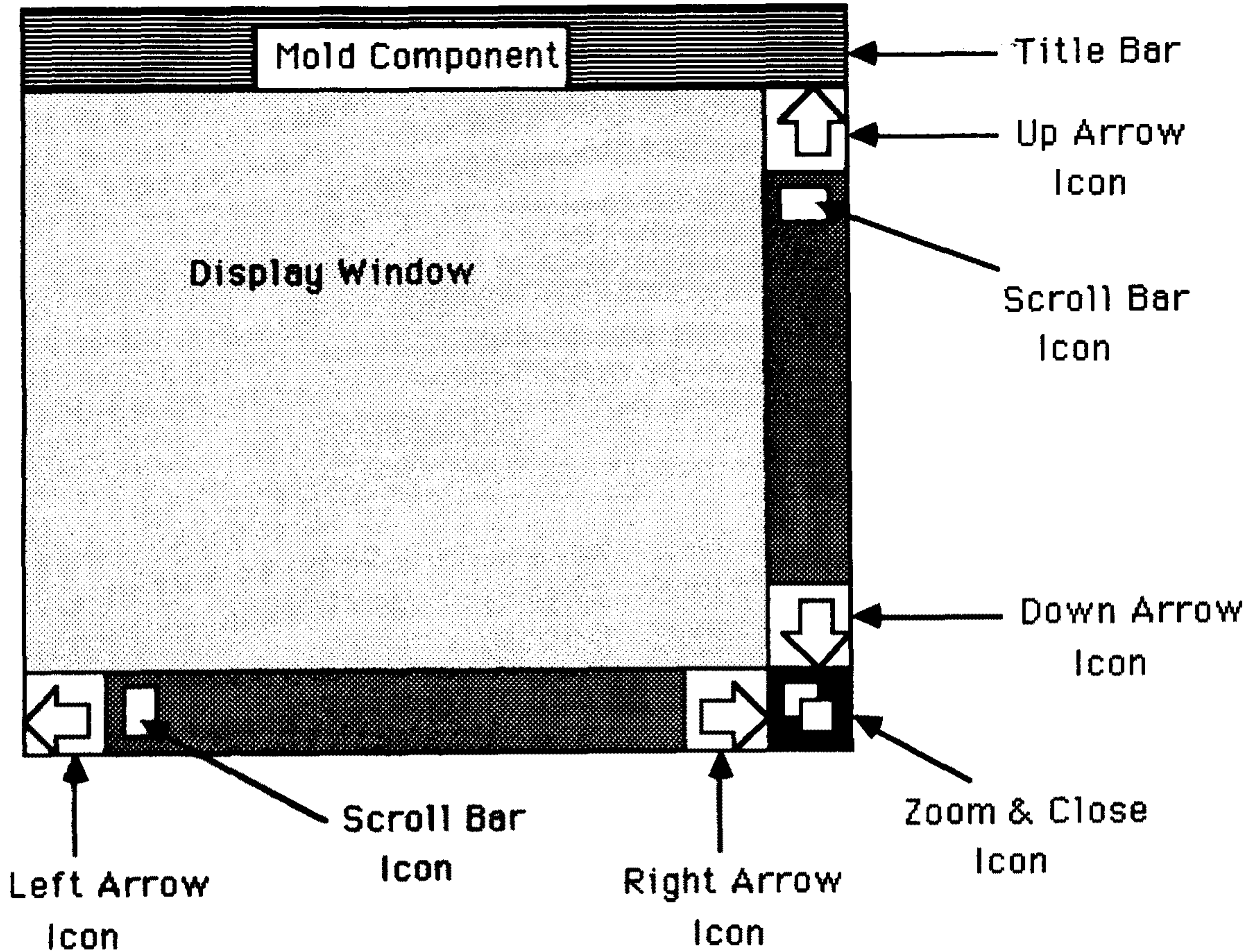
<그림 2-2> Pull-Down Menu의 예

3. Scroll Window & Zoom

Scroll Window 및 Zoom기능은 하나의 Window로 표현할 수 없는 큰 화면에 대해 좌우 Scroll과 Zoom에 의한 전체보기를 위하여 개발되었다. 플랜트 통제시스템의 몇개의 모듈에서는 커다란 그래픽화면이 요구되는 경우가 있다. 따라서 주어진 화면에 출력하기에는 부적합하고 설사 화면에 다 표시하더라도 사용자가 알아보기에는 힘들 정도로 그래픽요소들이 작게 나타나는 경우가 발생하기 때문에 평상시에는 전체 화면의 일부분만을 보여주다가 사용자가 원할 경우 전체화면을 보여줄 수 있도록 하는 기능을 가진 Tool을 개발하게 된 것이다.

Macintosh에서의 Utility에서 사용되는 화면(에를들어 NISUS, MacDraw, ...)이라든지, DOS에서 사용되는 Utility중의 하나인 MicroSoft WINDOW 프로그램에서 보여주는 그래픽 화면에서 아이디어를 얻어 개발한 Window Scroll을 위한 Tool은 사용자가 Mouse를 이용하여 Window의 가장자리에 위치한 화살표 Icon을 Click하면 화면이 원하는 위치로 움직이도록 하였고, 한편으로 우측하부에 위치한 Zoom & Close Icon을 왼쪽 Button으로 Click 하면 단계적으로 전체 화면을 보이도록 하고, 오른쪽 Button으로 Click 하면 Scroll Window에서의 작업을 종료하고 화면을 지우도록 개발하였다. 또한 화살표 Icon 사이의 임의의 부분을 Click하거나 Scroll Bar Icon을 Drag 하면 자동적으로 정확한 위치를 계산하여 그 부분을 화면에 나타내 준다. <그림 2-3>에 Scroll

Window에 사용되는 각각의 Icon들을 나타내었다.

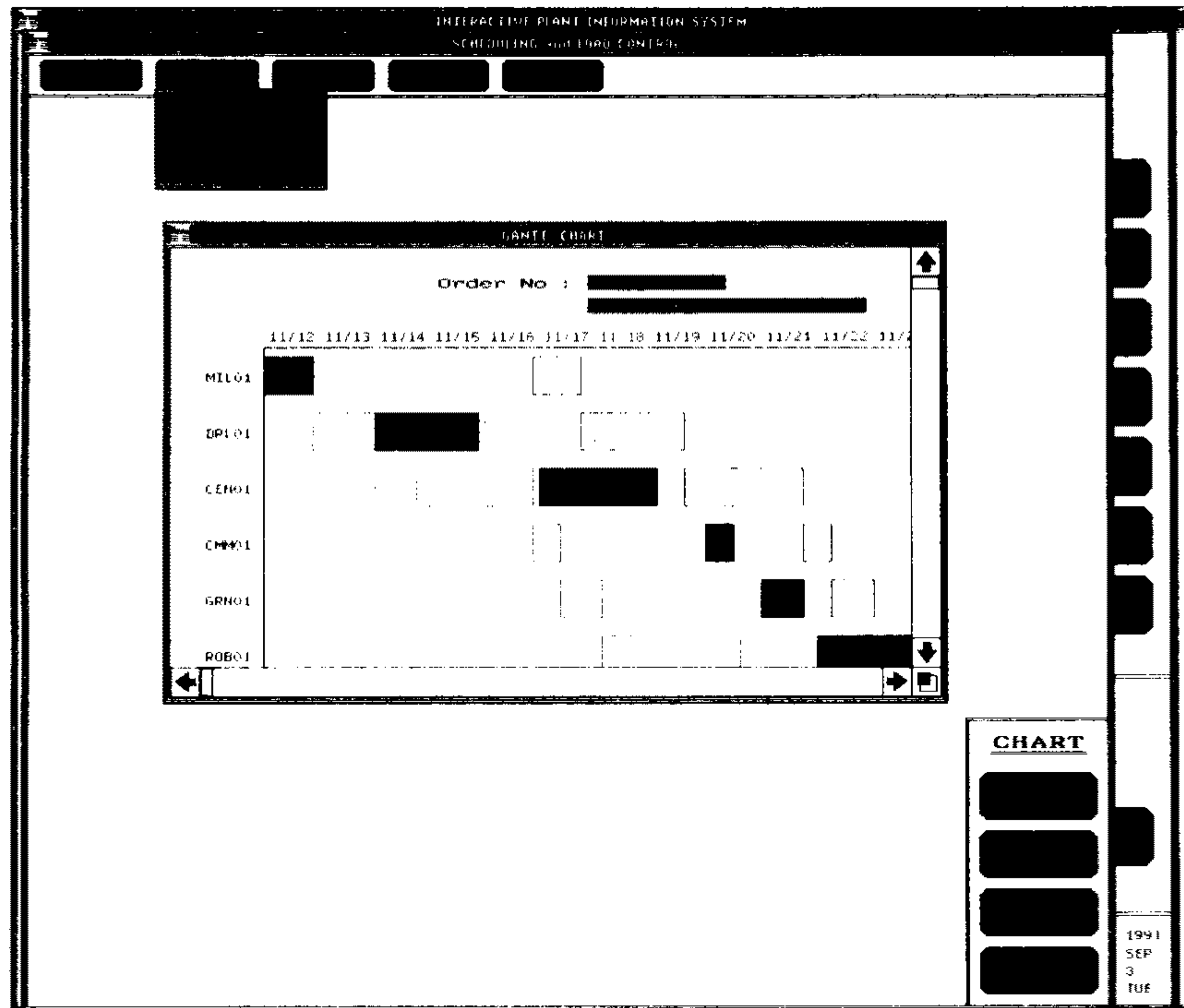


<그림 2-3> Scroll Window에 사용되는 각각의 Icon

Scroll Window을 위해서 개발된 프로그램은 기본적으로 두가지 작업화면을 제어한다. 하나는 각각의 Icon들이 있는 화면이고, 또다른 하나는 실지 그래픽이 출력되는 화면으로서 프로그래머가 작성한 가상화면(Virtual Display)상에 임의의 크기를 가진 출력화면(Display Window)이 그것이며 Window의 바깥부분에 있는

경계선을 제거한 후 Icon이 있는 Window위에 Overlap시켜 마치 하나의 화면에서 작업하는 것처럼 보이도록 만들었다. 출력화면은 대개 Window라 불리우며 전체의 내용중 일정부분만을 보여주는 창문이라 생각하면 된다. 동작방식은 사용자가 Mouse를 사용하여 Icon을 선택하면, Mouse를 누르고 있는 시간이라든가 위치등을 바탕으로 하여 가상화면(Virtual Display)중 사용자가 원하는 출력화면(Display Window)으로 Window를 움직여 화면에 나타내 준다. 본 프로그램을 위해서 프로그래머는 우선 화면에 출력하고자 하는 작업 화면을 가상 화면(Virtual Display)상에 구성한뒤 각각의 윈도우의 ID Number, 윈도우 Title, Pixel 좌표값, 윈도우 좌표값 그리고 윈도우가 위치할 좌표등을 제공하면 된다.

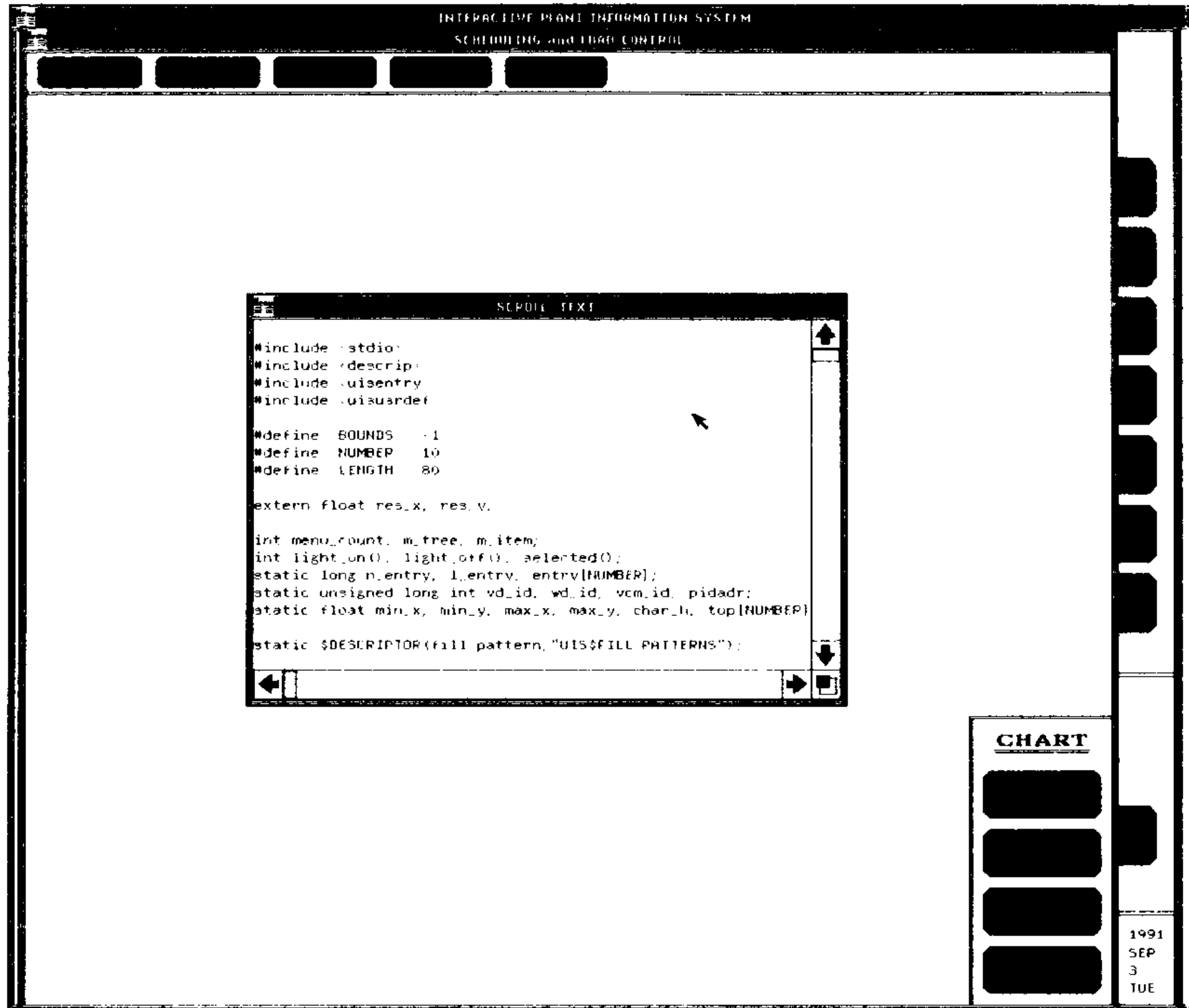
현재 응용된 부분은 Gantt Chart나 Load Profile등 시간축을 갖는 화면의 출력에 응용되었다. Gantt Chart의 경우 보통 3개월(약 100일) 정도를 가상화면(Virtual Display)상에 그리게 되는데 실제로 Scroll Window을 이용하여 화면에 출력되는 정도는 2주정도만 보여주게 된다. 물론 필요한 경우 가상화면에 그린 전체화면도 출력이 가능하다. 한편 PDA 모듈등에서는 현장의 Monitoring을 위해 전체 공장의 구성도를 가상화면에 그린후 Scroll Window을 이용하여 보다 편리한 PDA 모듈을 실행하려 하고있다. Scroll Window 및 Zoom에 대한 예는 <그림 2-4>에 나타낸다.



<그림 2-4> Scroll Window 및 Zoom에 대한 예

4. Scroll Text

플랜트 통제시스템에서 다루는 그래픽 요소중에는 Text File 혹은 Record Stream등이 있다. 이들 요소들은 그 크기에 있어서 매우 가변적이며 이를 화면에 표시하기 위해서는 앞에서 설명한 Scroll Window 기능과는 상이한 과정을 거친다. 즉, Icon을 이용한 조작방법은 거의 유사하지만 이를 화면에 나타내고 Scroll하는 과정은 전혀 다르다. 따라서 Scroll Window과는 달리 Text화면을 표시해주는 또다른 Tool이 필요하게 되었다.



<그림 2-5> Scroll Text의 예

Text 화면을 위해서 현재 개발된 Tool은 Scroll Text라고 명명되는데, Text의 크기에 관계없이 원하는 내용 모두를 효과적으로 화면에 나타내어 준다. Scroll Text을 위해서 개발된 프로그램은 외형은 Scroll Window기능에서 사용되는 화면과 동일하나 기능면에서 차이를 보인다. 즉 화면 출력에 있어서 역시 두가지 작업 화면을 제어하는데 하나는 각각의 Icon들이 있는 화면이고, 또다른 하나는 실지 Text가 출력되는 화면으로서 화면의 크기는 일정하다. 화면에 출력되는 데이터들은 Memory Allocation기법을 사용하여 컴퓨터의 메모리상에 기억되어 있어 사용자의 선택에 따라

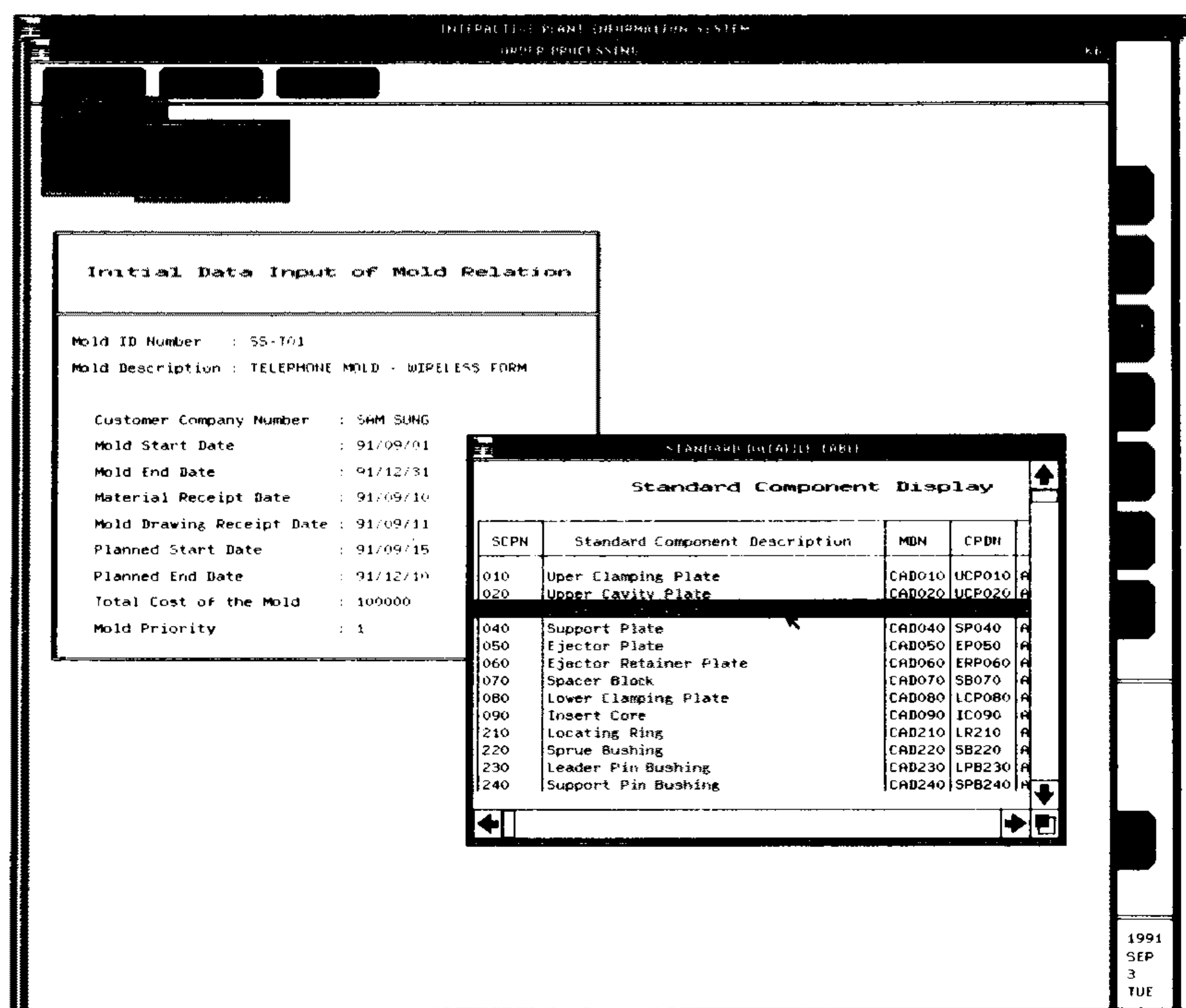
서 그때그때 불필요한 부분은 지우고 다시 그려나가는 방식을 사용한다. 동작방식은 사용자가 Mouse를 사용하여 Icon을 선택하면, Mouse 를 누르고 있는 시간이라든가 위치등을 바탕으로하여 메모리에 기억되어있는 자료중 사용자가 원하는 출력데이터를 찾아내어 현재 화면은 지우고 새로운 데이터로 구성된 화면을 그려낸다. 본 프로그램을 위해서 프로그래머는 우선 화면에 출력하고자 하는 데이터들을 Memory Allocation기법을 이용하여 컴퓨터의 메모리상에 저장한후 각각의 윈도우의 ID Number, 윈도우 Title, Pixel 좌표값, 윈도우 좌표값 그리고 윈도우가 위치할 좌표등을 제공하면 된다. Scroll Text에 대한 예는 <그림 2-5>에 나타내었다. Scroll Text의 종료는 Zoom & Close Icon을 선택하면 된다.

5. List Box

한편 화면에 나타나는 요소중에서 각종 데이터를 표시하는 Table을 나타내는 경우또한 플랜트 통제시스템에서는 필요하다. 그런데 Table출력에서는 정보의 시각적 출력이라는 중요한 기능 이외에 사용자가 이들 내용중에서 어떠한 정보를 추출하고자 하는 또하나의 목적이 있기 때문에 이를 지원하기 위한 선택기능이 필요하다. 따라서 Scroll Text의 기능중 각각의 데이터 Field에 대한 선택기능이 추가된 Tool을 List Box라 명명하였다.

List Box는 화면구성과 기능면에서 Text Scroll과 동일하며 데이터 추출을 위한 기능과 List Box내의 선택된 위치에 데이터

삽입을 위한 데이터 입력기능이 추가되었다. 각종 Field의 선택 방법은 Pull-Down Menu에서의 선택방법과 동일하다. 즉 사용자가 추출하려는 데이터가 있는 부분을 추적하여 화면에 출력한뒤 Mouse를 이용하여 원하는 데이터를 선택할 수 있으며 프로그래머는 이들 추출된 데이터를 바탕으로 적절한 작업을 시도하면 된다. 개발된 프로그램은 Macintosh나 MicroSoft WINDOW에서 지원하는 List Box의 기능만큼은 지원되지 않지만 그러한 기능을 염두에 두고 개발되었다.



<그림 2-6> List Box의 예

현재 응용된 부분은 작업관리 모듈에서 수주에 따른 작업

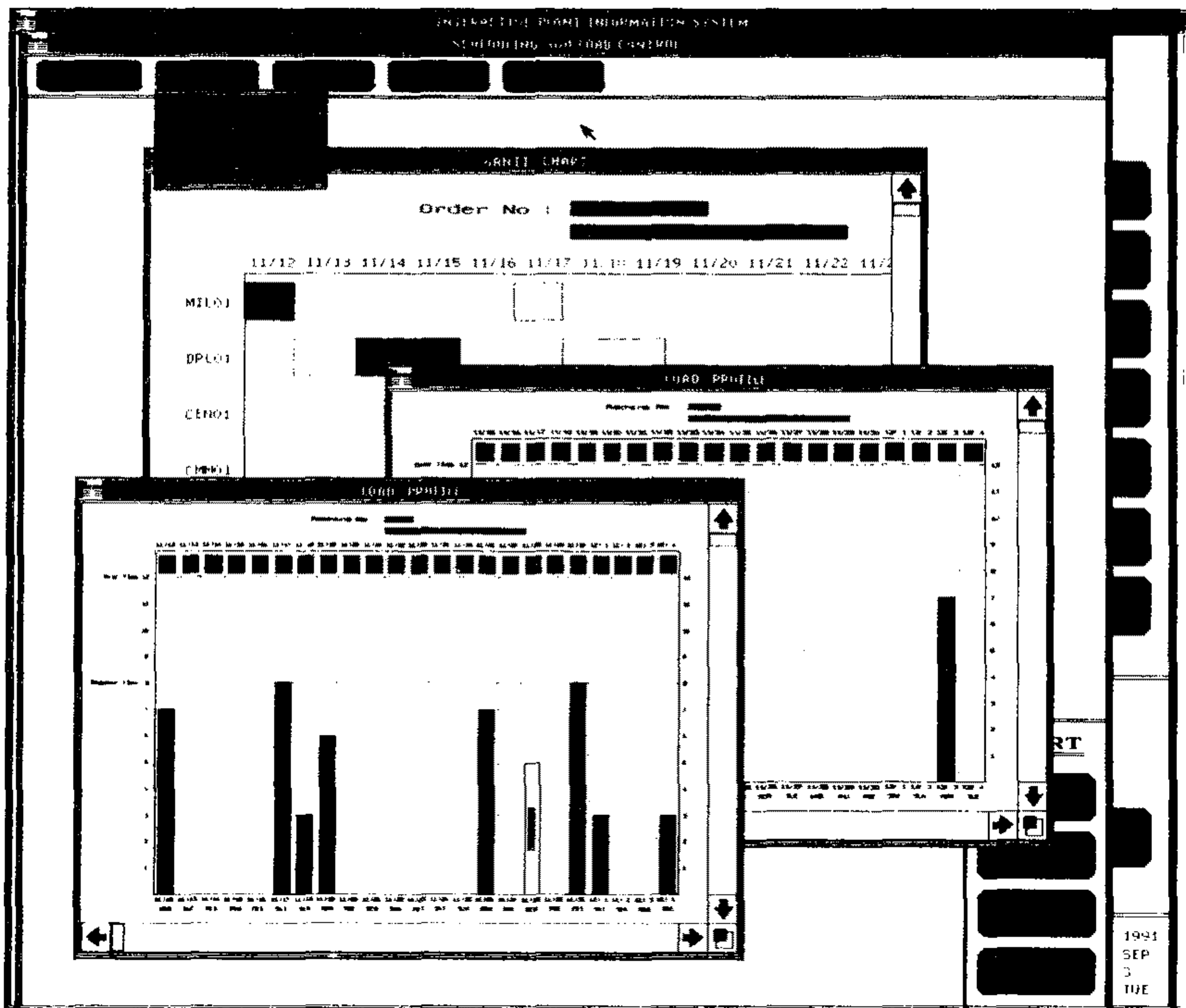
Order의 선택시 금형부품의 선택 혹은 각각의 Order에 대한 Operation의 선택 및 삽입, 삭제등에 이용되었으며 그밖에 NC Program 관리 모듈이라든지 PDA Monitoring 모듈등에 응용될 예정이다. List Box를 위한 예는 <그림 2-6>에서 설명한다.

6. Multi Window

Multi Window는 둘 이상의 그래픽 화면등을 보려할 때 필요한 Tool로 개발되었다. 즉 여러가지의 그래픽 Window가 동시에 출력되어 있는경우 사용자가 어느 한 Window에서 작업을 하다가 다른 Window로 작업영역을 옮기고자 할경우 컴퓨터가 현재의 작업 Window에 대한 정보(예를들어 작업 Window ID Number, Attribute, Display 위치, 현재 화면에 출력되어있는 그래픽요소들의 위치정보, ...)등을 놓치지 않고 추적해 나갈수 있도록 지원해주고자 작성되었다. 현재 VAX-Station에는 Process생성을 위한 기능은 있으나 위에 설명한 기능은 지원하지 않고 있기때문에 이를위한 Tool을 개발 하였다.

플랜트 통제시스템에서 요구되는 Multi Window 기능으로는 다음과 같은것들이 있다. 작업계획 및 제어 모듈에서는 둘이상의 Load Profile이나 또다른 Gantt Chart등을 보고싶은 경우 <그림 2-7>과 같이 화면에 출력되는데 Gantt Chart상에서 사용자가 임의의 위치를 검색하다, Load Profile의 내용을 찾고자 할때 각각의 Window사이를 왕래해야할 필요가 있으며 이 경우 Multi

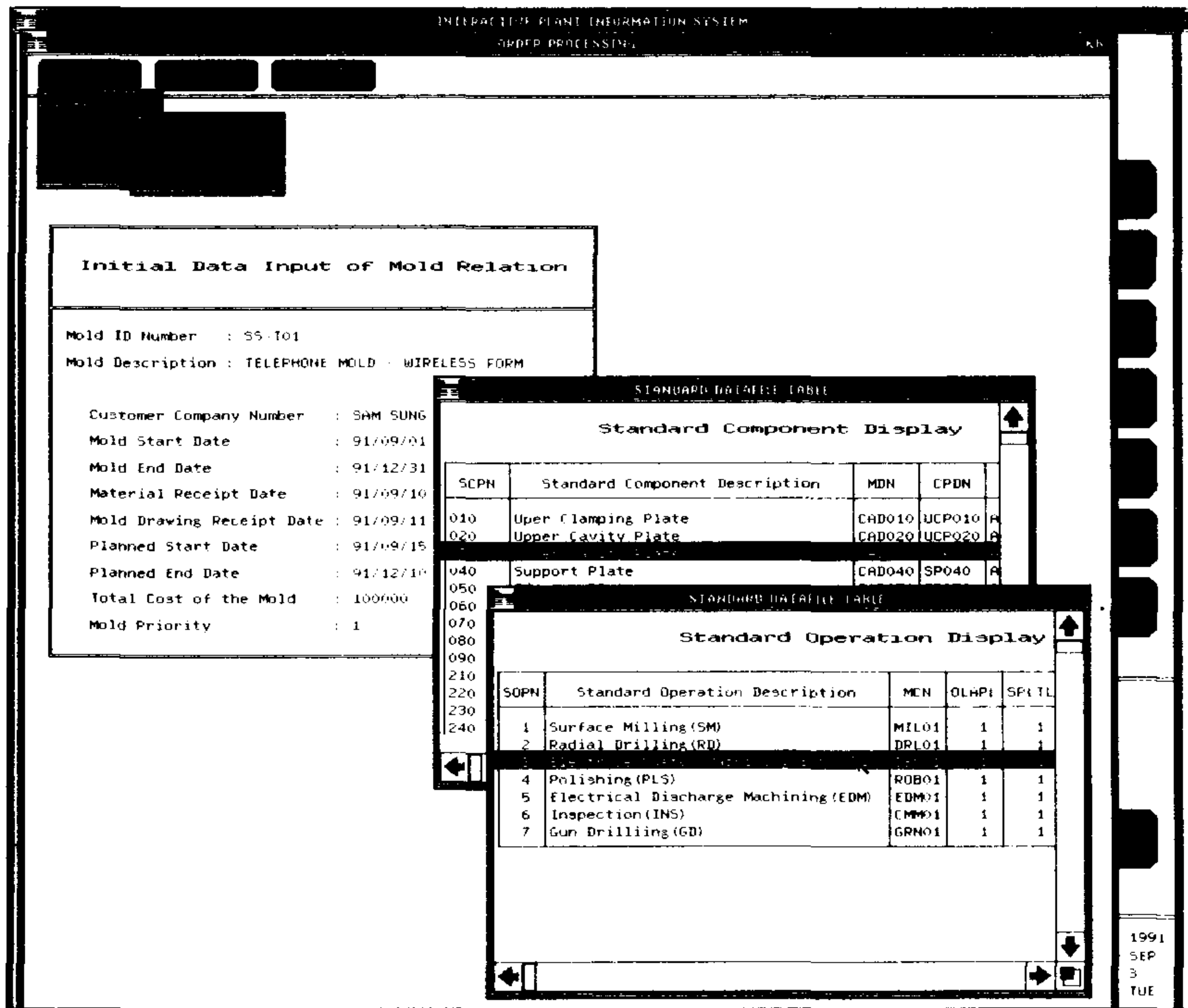
Window기법을 사용한다. 한편 작업관리 모듈에서는 Order에 대한 자료를 나타내는 List Box와 Operation에 관한 내용을 출력하는 List Box등을 동시에 화면에 출력되는 경우가 필요하며 <그림 2-8>과 같이 나타난다. 이밖에도 여러 모듈에서 충분한 정보의 제공을 목적으로 하는 여러개의 Window를 화면에 출력하여 사용하는 Multi Window의 기법이 필요하며 이를 적절히 사용해야 한다.



<그림 2-7> Multi Window 의 예 - 1

Multi Window에서 사용되는 Algorithm은 비교적 간단하다. 우선 Window가 생성되면 Memory Allocation에 의해 생성된 Window

의 정보가 저장될 기억장소를 만들고 Mouse의 Click으로 다른 Window로 이동하는 경우 선택된 Window의 정보들이 Memory Allocation된 기억장소에 새로이 Update 되며 이러한 동작이 반복적으로 일어나 언제나 사용자가 원하는 Window로의 추적이 가능하다. 현재 Multi Window는 n개까지 출력이 가능하나 VAX-Station에서의 Speed관계로 2 ~ 3개 정도가 적절하다.



<그림 2-8> Multi Window 의 예 - 2

7. Error Check & Message Display

여러가지 소프트웨어를 사용하는 도중에 있어서 가장 많이 경

협하는 사안중의 하나에 사용자의 실수 혹은 소프트웨어 자체내의 이유등으로 인하여 당하는 에러가 있다. 특히 소프트웨어 내부의 부주의로 인해 발생하는 에러는 충분히 제거될 수 있으나 프로그램의 개발당시 프로그래머 혹은 개발자의 경시에 의해 방치되는 경우가 많다.

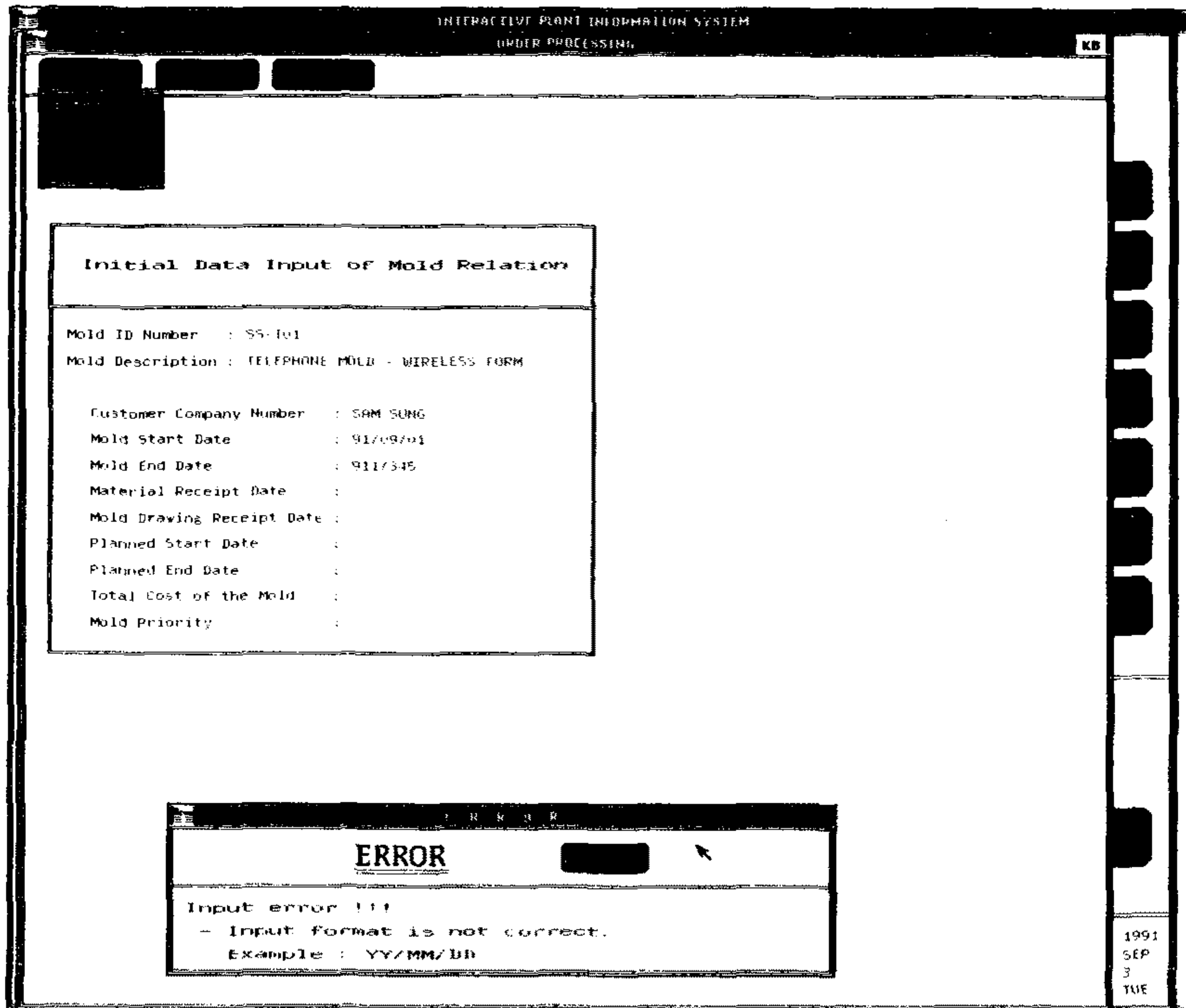
플랜트 통제시스템에서는 여러가지 데이터를 입력하는 부분에서 에러가 발생할 수 있는 소지가 가장 많은데, 이러한 에러를 제거하기 위하여는 입력된 데이터에 대하여 검증이 필요하다. 특히 Rdb를 사용하는데 있어서 저장될 데이터의 검증은 직접 Rdb를 이용하여 Error Check를 할수도 있으나 Rdb를 사용하는데 속도등의 문제로 여기서의 Check는 Performance를 떨어뜨린다. 따라서 직접적인 입력 또는 프로그램의 수행시 Error Check기능이 필요하며 이러한 요구에 부응하고자 Error Check를 전담하는 Tool을 개발하게 되었다. 결과적으로 Error Free상태의 데이터만이 Rdb에 저장되도록 프로그램을 개발하려 하였다.

Error Chaek에서 제공되는 기능으로는 Rdb 혹은 File등에 입출력을 위한 데이터의 입력시 다음과 같은 내용이 검증된다. 우선 입력되는 데이터의 Spelling Check를 비롯하여, 주어진 입력 Format에 따라 정수, String, 날짜, 혹은 시간에 대한 Format과 일치하는지에 대한 검증이 이루어 진다. 특히 날짜와 시간에 대한 검증은 VAX-Station에서 사용되는 몇가지 기능을 프로그램화하여 날짜의 경우, 입력되는 데이터의 범위를 검사하여 현재 사용가능한 날짜인지를 검사하고, 시간의 경우도 사용가능한 시간

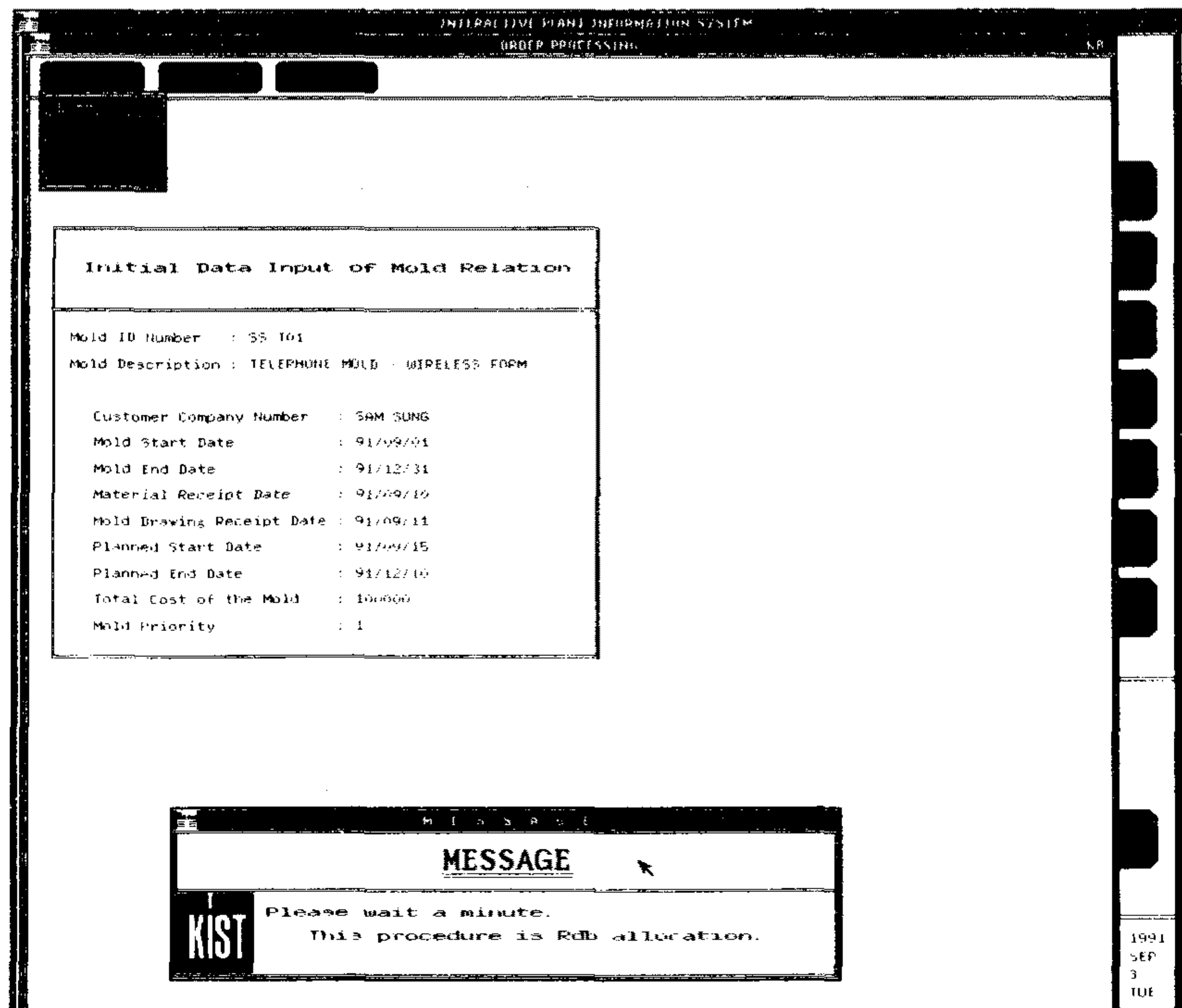
데이터인지를 검사하는등 데이터의 정확도를 높이기위한 검증을 해낸다. 예를들면 1991년 9월 40일 이라는 날짜의 데이터가 입력 될 경우 입력 데이터가 부적합하다는 Error Message를 출력하는 것이다. 따라서 데이터의 입력시 에러가 발생한 경우 사용자에게 적절한 대응을하도록 Error Message를 다양화하여 정확한 데이터의 입력를 유도하도록 하였다.

한편, 이밖에도 느린 수행속도등과 같은 시스템적인 이유등으로 사용자에게 여러 Information을 주어야 하는 경우가 많은데 각각의 경우에 적절한 Message기능을 첨가하여 사용자로 하여금 적절한 대응을 하도록 유도하는 기능이 필요하다. 이 기능또한 전체 모듈에서 필요하기에 본 플랜트 통제시스템을 유지하는데 사용되는 Tool로서 개발하였다.

Message 출력에서 담당하는 역할은 본 소프트웨어를 실행해 가는 과정에서 발생가능한 모든 잘못된 상황에 대하여 사용자에게 적절한 선택의 기회를 부여함과 동시에 시스템 자체의 Performance를 높이려는 관점에서 개발되었다. 즉 Rdb상의 데이터를 컴퓨터의 메모리상으로 추출해내는 Memory Allocation등과 같은 시간이 소요되는 작업에는 대기하라는 Message를 화면에 출력하고, 데이터의 입력이 끝난 경우에는 정확한 데이터가 입력되었는지 확인(Comfirm)하는 Message를 출력하거나, 어떤 이유에서 Error가 발생한경우 Error Message를 출력한다든지 하는등의 작업이 그것이다. 이러한 Error Chaek와 Message기능을 나타내는 예는 <그림 2-9>과 <그림 2-10>에 보여진다.



<그림 2-9> Error Check의 예

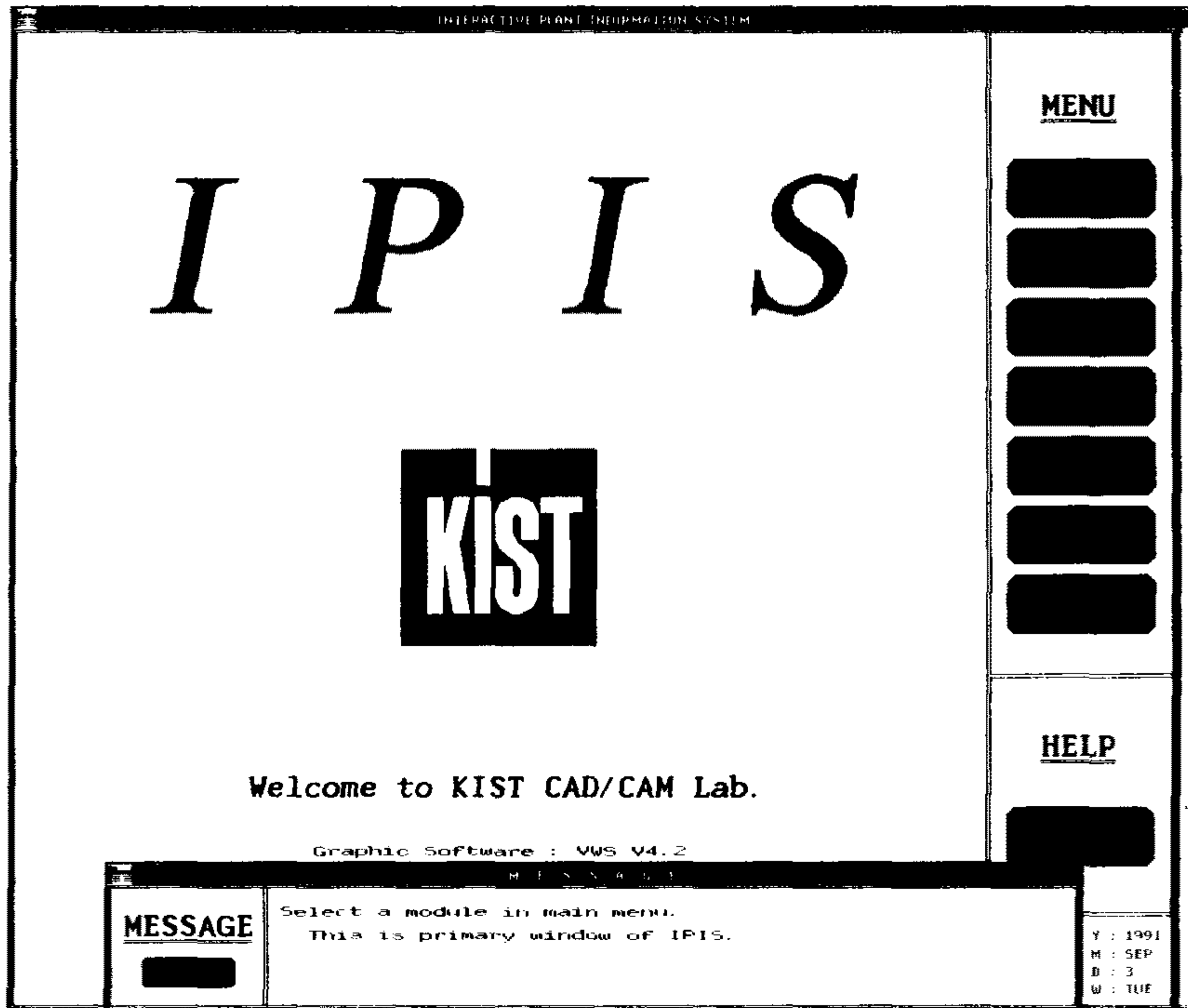


<그림 2-10> Message 기능의 예

8. On-line Help

플랜트 통제시스템과 같은 방대한 소프트웨어를 사용하는 사용자에게 있어서는 언제 어느때라도 시스템 운용에 대한 도움이 필요한 경우가 많다. 이를 위해서는 사용자가 원하는 어느때라도 도움요청에 대한 적절한 대응이 필요하게 되는데 이러한 기능을 지원하기 위해 개발된 Tool이 On-line Help기능이다.

Help기능을 위한 Help Icon이 초기화면 영역의 우측하부에 위치하기 때문에 이를 지원하기 위해 각각의 모듈이 표시되는 화면 영역은 본 플랜트 통제시스템이 시작되는 초기화면 영역보다 조금 작게 디자인 되었다. 따라서 사용자가 원할경우 언제든지 초기화면 영역에 있는 Help Icon을 이용하여 Help기능을 사용할 수 있으며, 현재화면에서는 어떠한 작업을 해야하는지 등에대한 Message를 출력한다. Help기능에 대한 예는 <그림 2-11>에 표시한다.



<그림 2-11> Help 기능의 예

제 3 장 작업관리 (Order Processing)

제 1 절 개요

1. 작업관리의 역할

작업관리에서는 금형의 수주에 따른 대일정 계획 및 증일정 계획의 수립에 필요한 대략적인 여러 자료들의 초기입력을 담당한다. 즉 각각의 일정계획에 따라서 수주받은 금형의 각 부품에 대한 작업 Order와 Operation들의 입력이 수행된다. 이 과정에서 먼저 설계도면요구일, 자재입고일, 생산시작일, 완료일 등을 입력 받으며, 각 Operation의 입력시 대략적인 추정 Leadtime으로부터 작업개시일과 작업종료일을 결정하고 각각을 해당 Machine에 할당하는 작업을 한다. 따라서 전반적인 데이터의 입출력이 수반됨과 아울러 이러한 데이터의 변경 및 삭제작업 또한 요구되기 때문에 궁극적으로는 일정계획에서 산출되는 여러 데이터 즉 작업 Order 및 Operation들에 대한 초기 Database를 구성하는 것이 주요 역할이다.

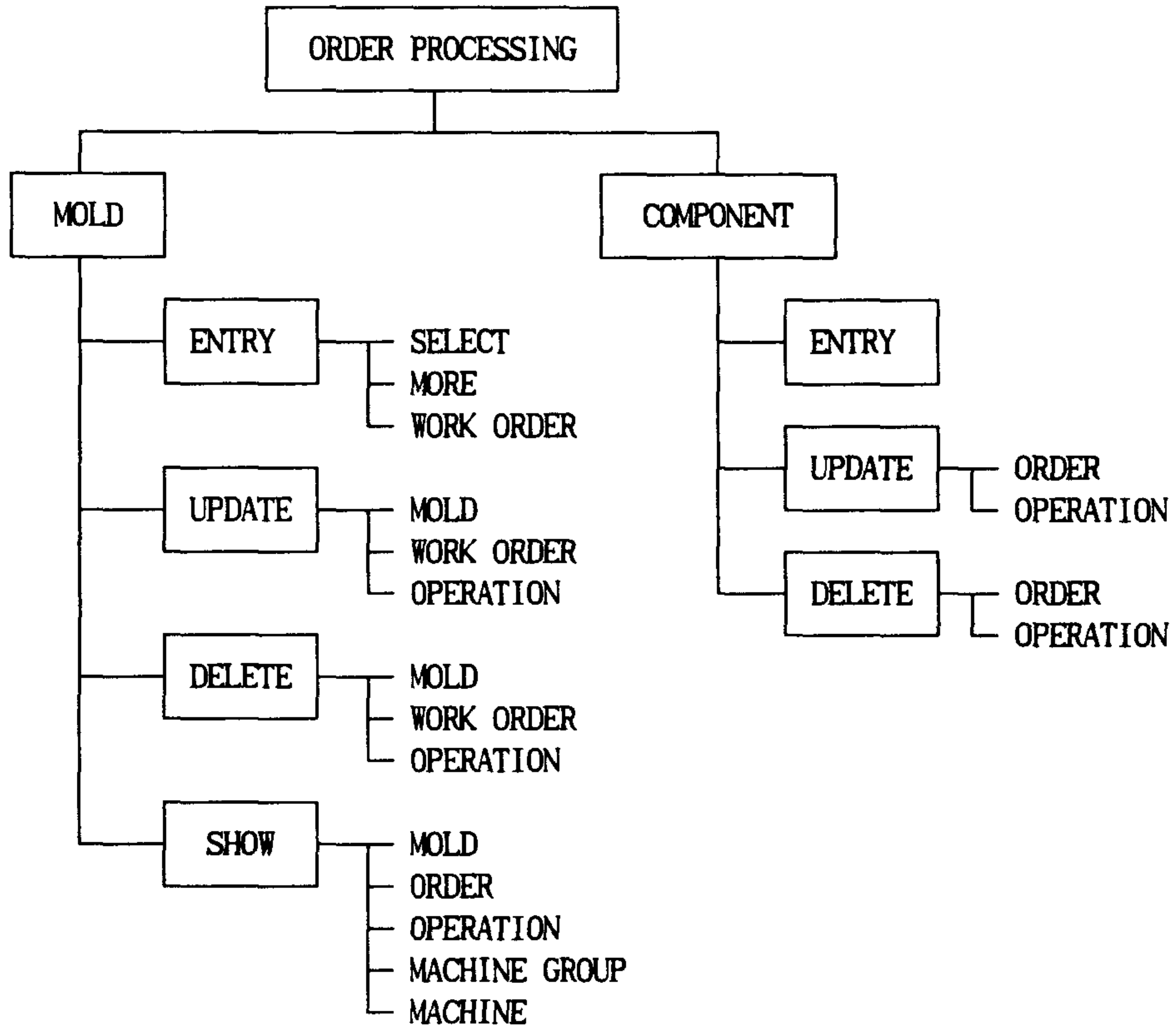
이렇게 작업관리에서 작성된 초기 Database를 바탕으로 하여 작업계획 및 제어 모듈에서 생산에 필요한 정확한 데이터로의 변화가 이루어지며 아울러 구성된 Database는 플랜트 통제시스템의 모든 모듈에서 사용가능하다. 따라서 각각의 모듈에서 Database

로 부터 추출해낸 데이터로 모듈특성에 맞는 작업을 행하게 된다.

2. 작업관리 모듈의 구성

앞에서 설명한 바와 같이 작업관리 모듈에서는 작업 Order의 수령, 입출력, 변경, 삭제 및 작업상태의 보고를 보여주는 모듈로서 주어진 Order 및 Operation에 대해 Database를 구성하는 주된 기능을 가지고 있다. 따라서 이 과정에서는 작업 Order에 대한 전반적인 입출력이 따르기 때문에 입력(Entry), 변경(Update), 삭제>Delete), 출력>Show)의 기능이 필요하다.

한편, 금형의 수주에 있어서는 금형 한벌 전체에 대한 수주의에도 특수한 금형부품에 대한 수주라든지 주문받은 금형 한벌중 어느 특정금형부품은 가공이 불가능하여 외부업체에게 하청을 맡기는 경우도 있게 마련이다. 따라서 이러한 데이터들을 한꺼번에 금형한벌 전체에 대한 데이터들과 합쳐서 관리하기보다는 금형한벌 전체에 대한 관리와 특수한 경우의 금형부품에 대한 자료를 관리하는 두가지 부분으로 나누어 다루는 것이 보다 효과적일 것이다. 이러한 관점에서 작업관리에서는 다음과 같은 두가지 Routine으로 구분하여 개발하였다. <그림 3-1>에 Routine의 구성에 대한 Structure를 나타내었다.



<그림 3-1> 작업관리 모듈의 구성

가. Mold Routine

Mold Routine은 금형 한벌에 대한 작업 Order를 관리하며 기본적으로 입력, 변경, 삭제, 출력등의 기능을 포함하고 있다. 즉 수주받은 금형에 대해 Order등에 대한 데이터를 각각의 Relation 별로 분류하여 효율적인 관리가 이루어 지도록 한다.

입력(Entry) Function에서는 수주받은 금형 한벌의 작업 Order와 Operation에 대한 초기입력이 이루어지며 이에 대한 초기 Database를 구성한다.

변경(Update) Function에서는 초기입력의해 구성된 Database에 오류가 있거나 혹은 Order의 변경에 따른 데이터의 변환이 요구되는 경우에 구축된 Database의 내용을 바꾸는 역할을 담당한다. Order의 변경에는 두가지가 있는데 하나는 Order 또는 Operation의 Recrod Attribute에 대한 변경이고, 다른 하나는 Order - Operation의 구조 변경이다.

삭제(Delete) Function에서는 이미 작업이 완료된 Order라든지 수주받은 Order의 취소가 있었다든지등의 경우에 있어서 더이상 Database의 자료로 관리할 필요가 없는 데이터들에 대해 이들을 지우는 작업을 담당한다. 필요없는 데이터가 많이 존재할 경우 시스템 전체의 운용속도도 저하될 뿐만아니라 관리하기도 불편하기 때문에 이들을 삭제하는 작업이 필요한 것이다.

출력(Show) Function에서는 현재 구성된 Database의 내용을 살펴볼 수 있도록 몇가지 기능을 두어 사용자의 요구에 따라 저장된 Database의 데이터들을 적절한 형태로 화면에 출력시켜주는 역할을 담당한다.

나. Component Routine

Component Routine은 하나의 금형부품에 대한 작업 Order를 관리하며 기본적으로 입력, 변경, 삭제등의 기능을 포함하고 있다. 주로 금형부품만을 주문받을 경우라든지 혹은 하청을 맡기는 금형부품들에 대한 관리가 목적이며, 그외에도 에도 적절한 역할

을 담당한다.

입력(Entry) Function에서는 수주받은 금형부품의 작업 Order와 Operation에 대한 입력이 이루어지며 초기입력에 대한 초기 Database를 구성한다.

그밖의 변경 Function이나 삭제 Function에서의 작업은 위에서 설명한 Mold Routine에 있는 작업과 동일한 역할을 담당한다. Component Routine에서는 출력 Function은 없는데 Mold Routine에 있는 출력 Function을 사용하면 동일한 효과를 얻을수 있다.

제 2 절 작업관리 모듈의 기능

현재 작업관리 모듈은 2장에서 설명한 여러가지 Tool들을 이용하여 새로운 화면구성을 통해 보다 User Friendly 하도록 유도해 나가고 있는 중이다. 이미 전년도에 작업관리 모듈은 어느정도 완성도를 보여주고 있었으며 현재는 시스템 통합을 위한 준비작업을 실행해 나가고 있다. 따라서 이미 개발된 여러가지 Tool들을 이용하여 프로그램의 수준을 높여가고 있다.

1. 금형 관리 (Mold)

가. 입력(Entry)

입력 Function을 시작하면 우선 금형 1벌의 대표가되는 정보

들에 대한 입력대기 상태가 된다. 사용자가 Keyboard를 사용하여 적절한 데이터를 입력하면 이들을 Rdb상에 저장하게 되고, 입력된 금형 ID Number에서 Mold Component를 선택할 수 있는 Code를 분리해 낸뒤 이를 바탕으로 표준 Mold Component들을 Rdb에서 검색한다. 한편 검색과정을 통해 추출된 데이터들은 Memory Allocation 기법을 사용하여 다음의 메뉴에서 사용가능하도록 컴퓨터의 메모리로 입력을 시키게 되며 마지막으로 입력모듈에 관련된 Pull-Down Menu가 화면에 출력된다.

(1) Select

선택(Select) Function을 선택하면 Memory Allocation 되어있는 금형부품들에 대한 정보가 List Box 형태로 화면에 출력되며, 주문받은 Order에 필요한 금형 부품들을 선택하게 된다. 이때 선택된 금형부품들 또한 컴퓨터의 메모리로 기억되며 다음에 설명할 Function에서 사용된다. 선택의 종료는 List Box내의 Zoom & Close Icon을 누르면 된다.

(2) More

추가(More) Function을 선택하면 선택 Function을 통해 메모리에 기억된 금형부품들외에 표준 금형부품에 없는 특별한 부품에 대한 정보를 입력할 수 있도록 새로운 List Box가 나타나며, 이 List Box에는 선택 Function에서 선택된 금형부품들이 출력되어 있다. List Box내의 하단부분에는 새로 입력될 금형부품에 대

한 입력대기 상태가 되어 있으며 주어진 Field에 데이터를 입력하면 된다.

(3) Work Order

작업오더(Work Order) Function을 선택하면 다음과 같은 순서로 작업을 진행할 수 있다. 우선 메모리에 기억된 각각의 금형부품들중 선택된 순서로 작업오더에 대한 정보를 입력받는다. 하나의 금형 부품에 대한 작업오더가 입력되면 그 금형부품을 가공하기 위한 표준 공정(Operation)을 Rdb로부터 검색하여 화면에 List Box 형태로 출력하며 동시에 Operation을 관리하기 위한 부 Menu가 화면에 나타난다. 이때 사용자는 Operation의 순서를 바꾼다든지, 새로운 Operation을 삽입한다든지, 혹은 필요없는 Operaiton을 삭제한다든지 하는 작업을 실행할 수 있다. 금형부품에 대한 Operation 순서를 결정한 후에는 이들 각각의 Operation들에 대한 정보를 순차적으로 입력받으며 입력된 정보들은 Rdb상에 저장된다.

나. 변경(Update)

변경(Update) Function에서는 초기입력에 의해 구성된 Database에 오류가 있거나 혹은 Order의 변경에 따른 데이터의 변환이 요구되는 경우에 구축된 Database의 내용을 바꾸는 역할을 담당한다. 변경 Function을 선택하면 변경에 필요한 작업을

수행할 수 있도록 Pull-Down Menu가 화면에 출력된다. 이 상태에서 사용자 원하는 Menu를 실행하면 된다.

(1) Mold

Mold Function을 선택하면 금형의 대표가되는 데이터들의 변경을 실행하게 된다. 우선 변경시키고자 하는 금형 ID Number를 입력하면 입력된 ID Number에 따른 각종 데이터를 Rdb에서 검색하여 화면에 출력시키게 되며 이때 사용자는 변경을 원하는 Field를 변경하면 된다. 변경된 데이터는 자동적으로 Rdb상에 Update 된다.

(2) Work Order

작업오더(Work Order) Function을 선택하면 금형 1벌의 Order들에 대한 변경을 가능하게 한다. Order가 속해있는 금형의 ID Number를 입력하면, 입력된 ID Number에 따른 Order들을 List Box로 출력시켜 주며 사용자는 변경을 원하는 Order를 선택한 후 나타나는 새로운 화면에서 선택된 Order에 대한 변경된 정보를 입력시키면 된다.

(3) Operation

공정(Operation) Function을 선택하면 각각의 금형부품들의 가공에 대한 Operaiton들의 변경을 행하게 된다. 금형부품의 Order ID Number를 입력하면, 현재 Operation들을 나타내 주는

List Box가 화면에 나타나며 사용자는 변경하고자 하는 Operation 혹은 삭제, 첨가하고자 하는 Operation들에 대한 정보를 입력하면 된다.

다. 삭제(Delete)

삭제(Delete) Function에서는 이미 작업이 완료된 Order라든지 수주받은 Order의 취소가 있었다든지 하는 등의 경우에 있어서 더이상 Database의 자료로 관리할 필요가 없는 데이터들에 대해 이들을 지우는 작업을 담당한다. 역시 삭제에 필요한 Pull-Down Menu가 화면에 출력되면 사용자가 원하는 작업을 실행할 수 있다.

(1) Mold

Mold Function에서는 금형한벌에 대한 모든 자료를 삭제한다. 금형 ID Number를 입력하면 Mold Relation의 각각의 Field에 대한 내용이 출력되며, 삭제할 것인가에 대한 확인(Confirm)에 사용자가 승인하면 금형 한벌에 대한 모든내용 즉 Order 및 Operation 등에 대한 내용을 추적해가며 삭제 작업이 이루어지게 된다.

(2) Work Order

작업오더(Work Order) Function에서는 금형부품에 대한 자료를 선택적으로 삭제한다. Order ID Number를 입력한뒤 Order

Relation의 내용이 출력되면 사용자의 승인에 따라 현재 입력된 Order 및 그 Order에 관계된 모든 Operation들을 삭제한다.

(3) Operation

공정(Operation) Function에서는 금형부품을 가공하기 위한 Operation에 관한 자료를 선택적으로 삭제하는 기능을 갖는다. 사용자가 Order ID Number를 입력하면 입력된 Order에 대한 Operation 들이 화면에 List Box형태로 출력되며 지우고자하는 Operation을 선택하면 확인 과정을 거쳐 삭제할 수 있다.

라. 출력(Show)

출력(Show) Function에서는 현재 구성된 Database의 내용을 살펴볼 수 있도록 몇가지 기능을 두어 사용자의 요구에 따라 저장된 Database의 데이터들을 적절한 형태로 화면에 출력시켜주는 역할을 담당한다. 즉 출력모듈에서 제공하는 Menu에 따라서 금형 전체(Mold), 오더(Order), 공정(Operation), 기계그룹(Machine Group), 기계(Machine)등에 관련된 데이터들을 사용자가 원하는 형태로의 출력이 가능하다. 따라서 기본적으로 Sorting기능외에 각각의 Menu에서 사용자가 원하는 특정 데이터 Stream을 구성하면, 이에 적합한 데이터들을 검색하여 화면에 출력시키는 기능을 갖는다.

데이터 Stream을 구성하기 위해서 검색하고자 하는 Relation

중 작업개시일, Order ID Number, Status등과 같은 검색 기준 Field를 사용자가 선택할 수 있도록 선택화면이 준비되어 있으며 이 선택화면에서 사용자가 출력형태를 정의하면 검색기준에 따라 Rdb를 검색하여 화면에 정확한 데이터들을 출력시키게 된다.

2. 금형부품 관리 (Component)

Component Routine은 하나의 금형부품에 대한 작업 Order를 관리하며 기본적으로 입력, 변경, 삭제등의 기능을 포함하고 있다. 주로 금형부품만을 주문받을 경우라든지 하청을 맡기는 금형부품들에 대한 관리가 목적이며 그외에도 적절한 역할을 담당한다.

Component Routine에서 처리하는 내용은 Mold Routine에서 설명한 기능과 거의 동일한 역할을 담당한다. 즉 금형부품의 수주에 따른 입력은 Entry Function에서, 금형부품의 변경은 Update Function에서 그리고 삭제는 Delete Function에서 작업을 수행하면된다. 기본적으로 처리하는 방식은 Mold Routine에서 실행하는 방식과 동일하므로 본 내용에는 설명을 생략한다.

3. 시스템 통합에 따른 추구방향

플랜트 통제시스템을 구축하기 위한 IPIS 프로그램은 현재 통합을 위한 과정을 거치고있는 중이다. 따라서 추진과정등에 대한

전반적인 내용을 살펴보면 다음과 같다.

우선 플랜트 통제시스템에 있는 전체모듈중에서 동일한 화면 구성을 통한 일관성을 추구하기 위해 필요한 몇가지 Tool은 거의 개발이 끝나 이를 각각의 모듈에 도입하여 사용하면 된다. 작업 관리 모듈은 현재 이들 Tool들을 바탕으로하여 화면의 재구성단계에 있으며 그밖에 다른 모듈에서도 점차적으로 도입하고 있다.

한편, Rdb상의 각각의 Field중 Status Code에 대한 부분은 그 사용방법을 대폭 수정하여 Bit연산 방식을 도입하였다. 예를들면 Machine Status Code의 경우 Wait, Idle, Operation, Completion, Breakdown등의 상황에 대해 각각의 상태를 Bit로 처리하여 특정한 상황 발생시 적절한 조치를 취할 수 있다. 작업관리 모듈에서는 초기 데이터입력시 이들 Status Code들을 내부적으로 초기화 시키고 있으며, 작업계획 및 제어, PDA, NC 프로그램관리등의 모듈에서 현 상태 파악등에 유용하게 사용되고 있다. 물론 각각의 상황변동들은 각각의 모듈에서 그때그때 Rdb상에 Update 된다.

시스템 통합시 간과해서는 안되는 부분또한 Rdb를 다루는 부분이다. 플랜트 통제시스템의 주된 기능이 Database를 이용하는 것이기 때문에 Rdb에 대한 내용이 가장중요한 부분을 차지한다고 할 수 있겠다. Database에 대한 Data Structure라든가, 구성방법 등에 대한 내용들은 전년도에 이미 밝힌바 있다. 그러나 Database를 구축하기 위하여 Rdb를 프로그램에 응용한 결과 처리 속도라든지, 시스템 관리의 일관성을 위한 측면에서 몇가지 부족

한 점이 있었기에 2절에서 설명한 Tool의 기능에 대한 개념을 도입하여 Database를 통합하여 일괄적으로 관리하고자 현재 Database만을 전담하는 모듈을 두고 있으며 이들의 기능들은 6절에서 설명할 것이다. 따라서 작업관리 모듈에서는 현재 이들 Rdb를 직접 Access하지는 않으며 6절에서 설명하는 몇가지 기능을 Tool로서 이용하여 프로그램화하고 있다. 다른 모듈에서도 동일한 방식으로 프로그램을 추진하고 있다.

제 4 장 작업계획 및 제어

제 1 절 작업계획 및 제어 모듈의 구성

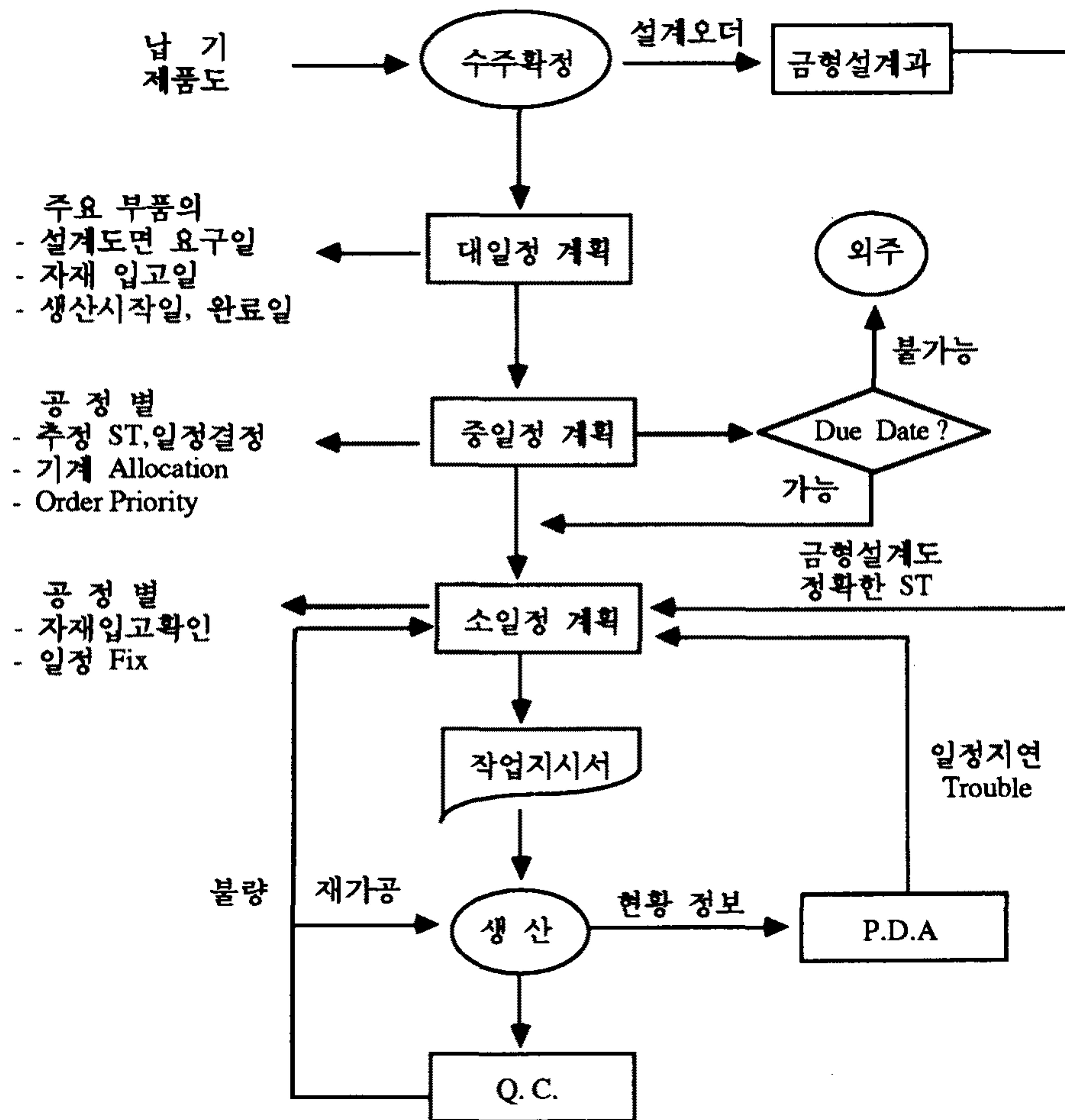
1. 금형공장의 일정관리

대부분의 금형 공장에서는 수주에서 출하까지 여러 단계를 거치며 각 단계에서는 다량의 정보를 입력 받아 많은 의사결정을 수행한다. 이러한 결정들은 그 하나하나가 다음 공정을 위한 또 다른 계획이 되며 계획과 실제 수행의 차이는 Feedback되어 새로운 계획 또는 기존 계획의 변경에 이용되어지도록 한다. 이러한 일련의 작업들은 각 회사의 특성과 제조품목에 따라 그 차이가 있으나 대체로 <그림 4-1>과 같은 Sequence를 거친다.

금형의 주문은 납기와 제품도의 주문으로부터 시작된다. 물론 주문이 있고나서도 현재의 작업 부하와 Schedule, 현장 설비를 토대로 그 주문을 감당할 수 있는지에 대한 검토가 선행되어야 한다. 이후에 상세히 설명되겠지만 본 Module에서는 작업장의 모든 기계에 대한 Schedule과 그날그날의 부하가 도식적으로 처리되어 나타날 것이다. 이를 보고 주문 담당부서에서는 주문을 받을런지에 대한 가·부를 결정한다.

수주가 이루어지면 금형설계과로 설계 오더가 나가고 이 부서에서는 주어진 제품도를 토대로 금형설계도를 작성한다. 이때 금

형의 특성을 고려한 각종 여유치수와 Rib등의 설계가 이루어진다. 또한 금형설계도가 완성되면 각 Operation에 대한 보다 정확한 Lead Time의 계산이 이루어진다.



<그림 4-1> 금형공장 일정관리의 흐름도

설계오더의 발주와 동시에 대일정계획이 이루어지는데 여기에서는

- 설계도면 요구일
- 자재 입고일
- 생산 시작일 · 완료일

의 3가지 Data가 결정된다. 설계도면 요구일은 금형설계 부서에서 금형설계도가 완성되어야 할 날짜이다. 금형 생산을 위해서는 자재의 준비가 필수적인데 제품도로부터 BOM(Bill of Material)을 구하고 이를 토대로 필요한 자재의 List를 만들어 자재부서에 그들이 필요한 날짜를 알려준다.

또한 생산 시작일과 완료일이 결정되는데 이는 작업부하와 금형의 납기일을 적절히 조화시키는 선에서 결정된다. 이러한 대일정계획은 금형 생산을 위한 가장 대강의 계획으로 수주와 동시에 수행될 수 있다.

대일정계획이 이루어진 후 이 결과는 제작부서로 넘어가 증일정 계획을 수립하는데 이용된다. 여기서는 금형 전체에 대해 생각하던 대일정계획과는 달리 금형의 각 부품별 공정에 이르기까지 세부계획이 수립되는데 이 역시 어느 정도 대강의 계획이다. 여기서의 출력으로는

- 각 부품별 Lead Time 수정
- 일정 결정 (Scheduling)
- Machine Allocation
- Order Priority 결정

- 외주 결정

의 4가지이다. 추정 Lead Time은 제품도로부터 어느 정도 금형 설계에 대한 식견이 있는 사람이 산출하는데 이는 각 부품의 대략적인 가공시간(Standard Time)과 공정(Operation)간의 운송시간(Transport Time)의 합이다. 이 시간과 현재의 작업 부하를 대조하며 일정계획이 수립되는데 이때 전방계획법 또는 후방계획법 등의 Scheduling기법을 동원하여 대략적인 일정을 잡는다. 또한, 이 과정에서 각 기계에 대한 작업 할당이 이루어지는데 이때에도 대략적인 Load Balancing만을 수행한다. 이상의 일정 계획 및 작업할당을 토대로 Order Priority를 결정한다. 이 Priority의 결정 Rule에도 여러가지가 있으나 대표적인 것으로 최소 여유 시간 규칙이 있는데, 이는 여유시간이 짧은것부터 처리되도록 한다. 이상의 과정을 밟아오는 동안 납기내에는 도저히 처리할 수 없는 부품에 대해서 외주를 주도하도록 한다.

이러한 중일정계획을 수행하는 이유는 수주후 금형설계도를 작성하여 정확한 Lead Time의 계산이 이루어지기까지 많은 시간이 소요되고 그 결과, 현장에는 부하가 할당되어 있음에도 불구하고 Database에도 저장되지 않아 부하율의 계산 및 현재의 부하를 토대로 한 차후의 계획에 오류를 유발하기 때문이다.

중일정계획의 결과와 금형설계과로부터 넘어오는 금형도 및 여기서 결정된 Lead Time등이 소일정계획의 Input이다. 이때의 수행 결과는

- 일정 Fix

- 작업 지시서 발급

의 2가지이다. 금형설계과로부터 나오는 금형설계 도면과 정확한 Lead Time은 이 단계에서 확실한 Scheduling이 이루어지도록 한다. 이때의 작업은 새로운 일정계획이 아니라 증일정계획에서 얻어진 Schedule의 수정 과정이다. 따라서 각 Order 및 Operation의 계획날짜(Planned Date)들이 약간씩의 조정을 거친 후 이 결과가 작업지시서의 형태로 해당되는 날짜에 현장으로 보내어진다. 물론 이때에도 도식적으로 처리되는 각 Order의 Gantt Chart 및 그 Order를 수행하는 각 Machine의 Load Profile이 의사결정자에게 시각적인 정보를 제공한다.

소일정계획의 결과로 나온 작업지시서는 현장담당자에게로 넘어가며 이 사람은 그날그날 내려오는 이 작업지시서를 토대로 기계가공을 수행한다. IPIS System의 PDA Module은 이 현장에서 현장담당자 등에 의해 수행되는데 현재의 작업현황, 즉 작업개시, 작업종료, 고장발생등의 정보를 수집하여 중앙에 위치한 Data Base로 전송한다. 뿐만 아니라 일정 지연이 발생한 경우와 불량 등의 결과도 중앙에 보고하여 차후의 새로운 계획에 반영되도록 한다.

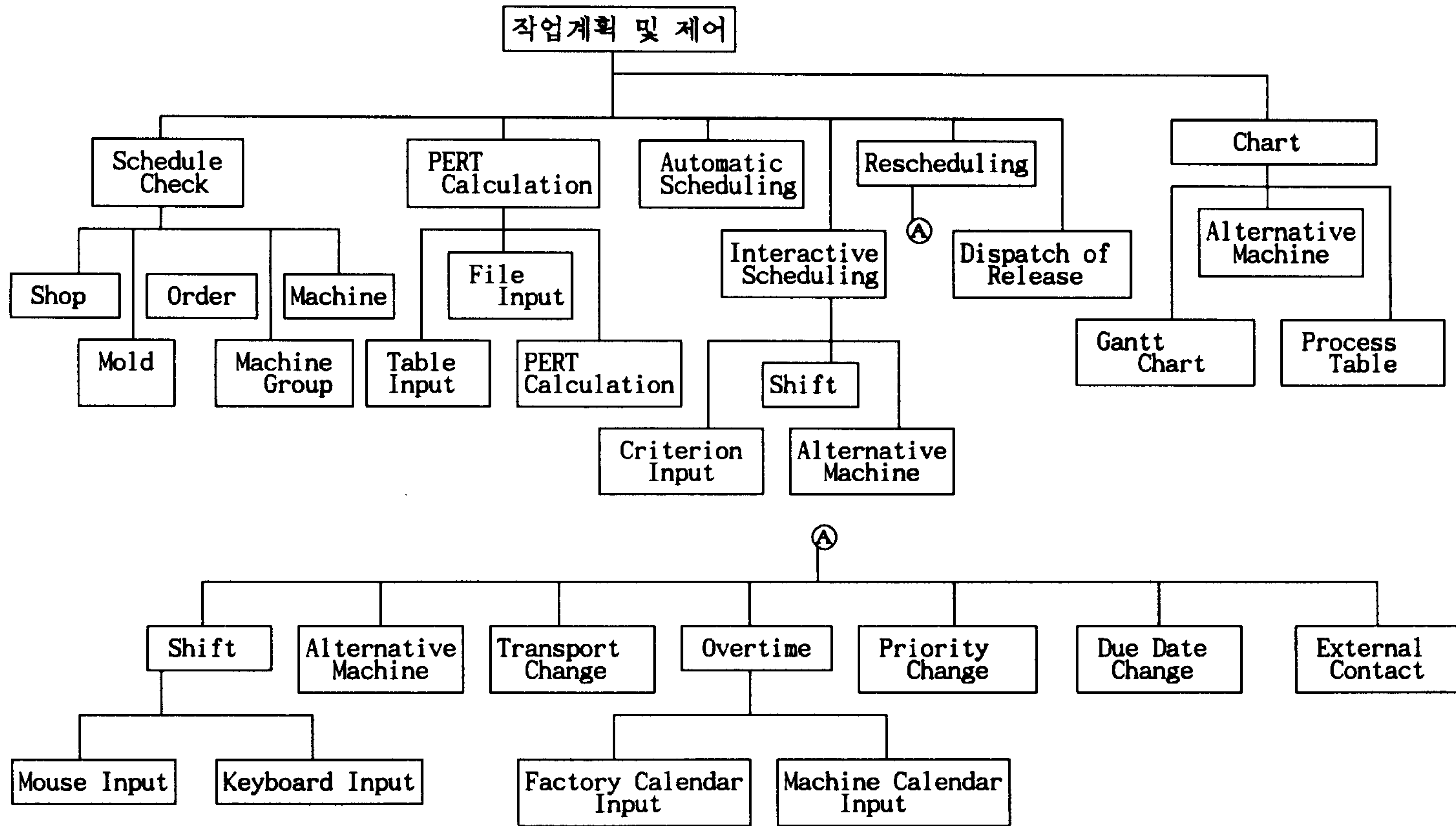
2. 모듈의 구성

앞서 살펴본 금형공장의 일정관리는 수주에서부터 현황정보 수집 및 실적관리에 이르기까지 금형생산의 전 과정을 총체적으

로 고찰한 것이다. 이 내용중에서 Scheduling에 관련된 부분만을 모아 놓은 것이 이 모듈의 내용이다.

먼저 Order에 대한 Planned Date 및 해당 Operation들의 Lead Time을 입력하여 PERT Calculation을 수행한다. 다음으로 주어진 Criterion에 따라 Order별로 하나씩 Scheduling을 행하고 동시에 Machine Allocation을 수행한다. 그 다음으로 필요에 따라 Rescheduling을 행하는데 그 방법에는 Shift, Alternative Machine, Transport Time 변경, Overtime 등 여러가지가 있는데 사용자는 이들을 이용하여 기존의 Schedule을 재계획할 수 있다. 이상으로 얻어진 Schedule에 대해서는 Schedule Check Routine의 수행으로 그 적합성 여부를 알아낼 수 있다. 끝으로 완성된 Schedule을 토대로 그날그날의 작업 지시서가 현장의 작업 반장 또는 작업자에게 직접 전달되어 진다.

금형공장 일정관리의 흐름도를 다시 한번 살펴 보면 대일정관리, 중일정관리, 그리고 소일정관리의 실행 Routine이 어디인가에 대한 의문을 가질 수 있다. 대일정관리, 중일정관리는 작업관리모듈에서 수행되어 진다. Order, Operation의 입력 및 Machine의 Allocation이 여기에서 이루어지는데 이때의 입력 근거는 수주 또는 계획생산 결정 등에서 얻어지는 경영정보와 제품도를 통한 추정 Data이다. 이들 중 일부는 작업관리 모듈에서 직접 입력할 수 있는 것도 있고 <그림 4-2>의 작업계획 및 제어 모듈의 지원을 받아야 하는 것도 있다.



<그림 4-2> 작업계획 및 제어 모듈의 VTOC

특히 추정 Lead Time을 이용한 Scheduling과 Machine Allocation은 작업계획 및 제어 모듈의 방법과 거의 동일하여 여기서 다른 Order의 Gantt Chart 또는 해당 Machine의 Load Profile등을 출력해 볼 수 있다. 작업계획 및 제어 모듈과 다른 점이 있다면 여기에서의 계획은 보다 대강의 것으로 그날그날의 실제 수행을 염두에 둔 것이라기 보다는 Load Balancing등을 통한 수행가능성에 그 목표를 둔다는 점이다

<그림 4-2>는 작업계획 및 제어 모듈의 VTOC(Visual Table of Contents)로 이 모듈의 Menu System과도 상통한다. 이들 중에서도 이 모듈의 근간을 이루는 것은 Automatic Scheduling, Interactive Scheduling, 그리고 Rescheduling의 세가지이다. Automatic Scheduling은 모든 Scheduling - Machine Allocation을 제외한 날짜에 관련된 모든 계획 - 과 Machine Allocation을 주어진 Algorithm에 따라 자동 수행하고자 하는 모듈이다. Interactive Scheduling은 기존의 현장 Schedule을 Base에 깔고 온 위에 새로이 Order의 Schedule을 주어진 Criterion에 따라 단순히 쌓아 올리는 계획 방법이다. 즉 새로운 Order의 각 Operation들이 해당 Machine의 현 Load 상태를 움직이지 않는 범위에서 전방계획법이면 Order의 시작일에 후방계획법이면 Order의 종료일에 맞추어 Load로 등록한다. 그리고 Rescheduling은 기존의 계획이 Rush Order, 작업 지연 등의 이유로 정상적인 수행이 불가능한 경우 재계획을 가능하게 하는 Routine이다.

제 2 절 작업계획 및 제어 모듈의 기능

1. Schedule Check

이미 결정된 Schedule에 대해서 과부하가 발생하지 않았는지, 주어진 납기를 넘기지 않았는지에 대한 Check가 수행 중 종종 필요할 것이다. IPIS(Interactive Plant Information System)는 그 명칭에서도 볼수 있듯이 대부분의 결정을 사용자가 대화식으로 결정한다. 따라서 그 결과는 Computational Optimization이 수행되지 않고 사용자의 의지에 따른 불균일한 부하의 분배가 이루어진다. 구체적으로 설명하면 한 기계에 정규 부하가 존재할때 이상적인 경우라면 기계가 가동되는 모든 시점에서 정규부하를 꼭 맞추도록 Scheduling이 이루어질 것이다. 그러나 사람이 하는 경우는 정규부하를 중심으로 과부하 또는 아부하 상태로 Scheduling되는 날이 많이 발생하게 되는데 특히 기계가 많고 작업장이 대형화 할수록 Human Factor에 의한 작업부하의 불균일은 증대될 것이다.

이러한 과부하 발생 이외에도 납기지연, 미등록 기계 또는 공구의 사용 등 Error요인은 여러가지이다. 이들을 정리하면

- Due Date Check
- Overload Check
- Order, Operation의 이중 등록 Check
- Operation Sequence Check

- 각종 Status Code Check
- Calendar와 현장 일정간의 비교
- 미등록 Machine이나 공구의 사용여부 Check

등으로 나타난다. 이러한 Check Point들의 수행 결과는 다양한 화면 출력으로 사용자에게 그 상태를 제시해 준다.

먼저 Error가 발생하면 제 2장에서 설명한 Error Message출력 기능에 의해 <그림 2-10>와 같은 Window가 Sound와 함께 나타난다. 사용자는 이 Message를 토대로 그 Error가 무엇인지를 인지하고 보다 직접적인 내용을 알기 위해서는 System의 보조 기능을 이용하여 확인한다. 예를 들면, Overload라는 Error가 발생한 경우 Message에 해당 기계와 발생 날짜가 동시에 출력된다. User는 이를 토대로 해당 기계의 Load Profile을 출력시켜 과부하 발생 정도를 시각적으로 알 수 있다.

가 Schedule 검증을 위한 Check Point

(1) Due Date Check

현장에서 흘러가는 계획의 단위는 그 출발점이 주로 Order의 Planned Date이다. Order의 Planned Date는 Order에서 수행해야 할 Operation들 중 맨처음 Operation의 Planned Start Date와 맨 마지막 Operation의 Planned End Date를 포함해야 한다. 즉, Operation들의 Lead Time을 모두 더한 것이 Order의 Lead Time을 초과해서는 안된다. 또한 Operation들 각각에 대해서도 Earlist

Start Date와 Latest End Date를 Over하지 않는 범위에서 Planned Date가 결정되어 있는지를 조사한다.

(2) Overload Check

현장의 각 기계별로 날짜 단위의 부하를 조사한다. 해당 Machine의 하루 정규부하가 8시간이라고 할때 이보다 많은 부하가 걸린 경우가 Overload가 되며 이 경우 Error 또는 Warning Message를 발송한다. Error인 경우는 System의 운영 방식에 따라 다를 수 있으나 여기서는 하루 24시간 부하를 초과한 때이고 정규부하보다 크고 24시간 이하인 경우는 Warning으로 처리한다.

(3) Order, Operation의 이중 등록 Check

Rdb를 이용하는 경우 Relation의 Primary Key로 등록된 Field에 대해서는 자동적으로 그 Duplicate를 Check한다. 그러나 Operation의 경우처럼 Primary Key가 없는 경우에는 그 이중 등록 여부를 Check해야 한다. Operation번호는 Sequential Number를 이용하는데 한 Order에 대해서 그 수행 순서를 유지하고 있는지에 대한 Check도 수행되어야 한다.

(4) Operation Sequence Check

Operation Sequence의 검증은 Program 내부에서 자동 처리되는 과정이 아니다. 이것을 수행하는 경우 주어진 Order의 Operation들이 순차적으로 화면에 출력되며 사용자는 이 결과를

직접 확인해야 한다. 출력 화면을 Close시키는 과정에서 Query가 나와 검증 결과로 받아 들인다. 이러한 특징 때문에 이 과정은 Automatic Check과정에서는 제외된다.

(5) 각종 Status Code Check

Status Code들에 대한 설명은 2장 2절에서 상세히 설명하였다. 이들 Status Code중 일부는 다른 Status Code와 서로 연관 관계를 갖는 경우가 있다. 즉, Operation Status Code 중 Begin과 End가 동시에 Set되어 있으면 Error 상태이다. 또, Order Status Code는 End이고 그 Operation들 중 어느 것의 Status Code가 Begin인 경우는 Error이다. 이처럼 Status Code들 간에도 복잡한 상관관계로 인해 Error를 유발할 우려가 많아 그 Check를 필요로 한다.

(6) Calendar와 현장 일정간의 비교

IPIS에는 Factory Calendar를 유지하는 기능이 있어 현장의 수행 일정이 보다 정확한 계획이 되도록 하였다. Factory Calendar는 Rest Time, Over Time, 그리고 휴일등을 관리하는데 이를 이용하여 휴일임에 불구하고 작업수행으로 계획된 일정에 대해서는 Error를 발생한다. 보다 구체적으로 설명하면 1월 1일의 경우는 신정 휴가일로 현장 모두가 쉬는데 이날 작업이 있는 것으로 일정이 잡히면 이 일정은 의미없는 것이 되고 만다. 이러한 오류를 방지하기 위해 이 기능을 수행한다.

(7) 미등록 Machine이나 공구의 사용여부 Check

등록되지 않은 Machine이나 공구가 작업지시서에 출력된다면 현장에서는 당황함과 함께 Schedule에 대한 불신을 표출할 것이다. 이를 방지하기 위해 Database 검색을 통해 등록되지 않은 Machine, 공구, Fixture가 있는지를 Check한다.

나. Schdule Check Routine의 Menu 구성

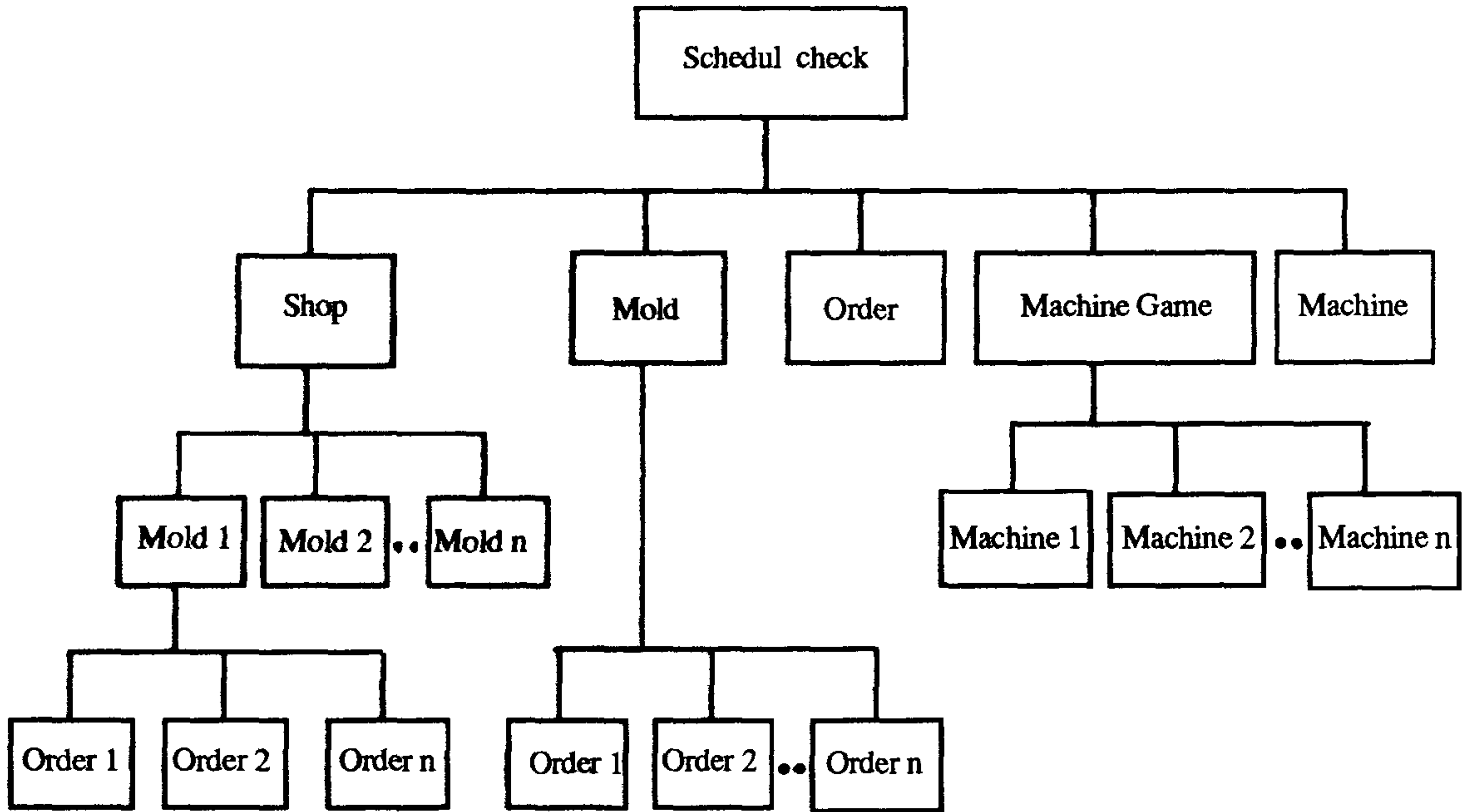
앞서 설명한 Check Point를 기준으로 Schedule에 대한 검증을 실시하는데 그 대상은

- Shop
- Mold
- Order
- Machine Group
- Machine

의 5가지로 <그림 4-3>의 VTOC와 같은 구성을 갖는다.

(1) Shop

Shop에 n개의 Mold가 있고 Mold마다 또 n개의 Order가 있다고 할때 그 모든 Order들에 대해 Schedule Check를 수행한다. 즉 현장 전체에 대해 Schedule을 검증하는 작업으로 시간과 노력이 많이 소요되므로 매월 또는 그 이상의 간격으로 수행하거나 돌발사고로 System의 Database를 재 구축한 경우 등에 이용한다.



<그림 4-3> Schedule Check Routine의 VTOC

(2) Mold

2장에서 설명한 것처럼 한 Mold는 여러개의 Component로 구성되고 그 각각은 Order로 등록된다. 따라서 금형 한벌 단위로 Schedule을 검증해야 하는 경우가 있는데 이때에 이를 이용한다.

(3) Order

Schedule 검증시 기본 단위는 항상 Order가 되며 한 Order만

의 Schedule 검증은 여기서 한다. 이때의 Check Point는 앞서 설명한 7가지이다.

(4) Machine Group

Order 관점이 아닌 Machine 관점의 Schedule 검증도 필요한데 이에 대한 기능이 여기서 수행된다. Parallel Machine들을 모아 Group으로 만들고 여기에 대한 전체 일정을 검증한다. 이때에는 주로 부하에 대한 Check가 이루어진다.

(5) Machine

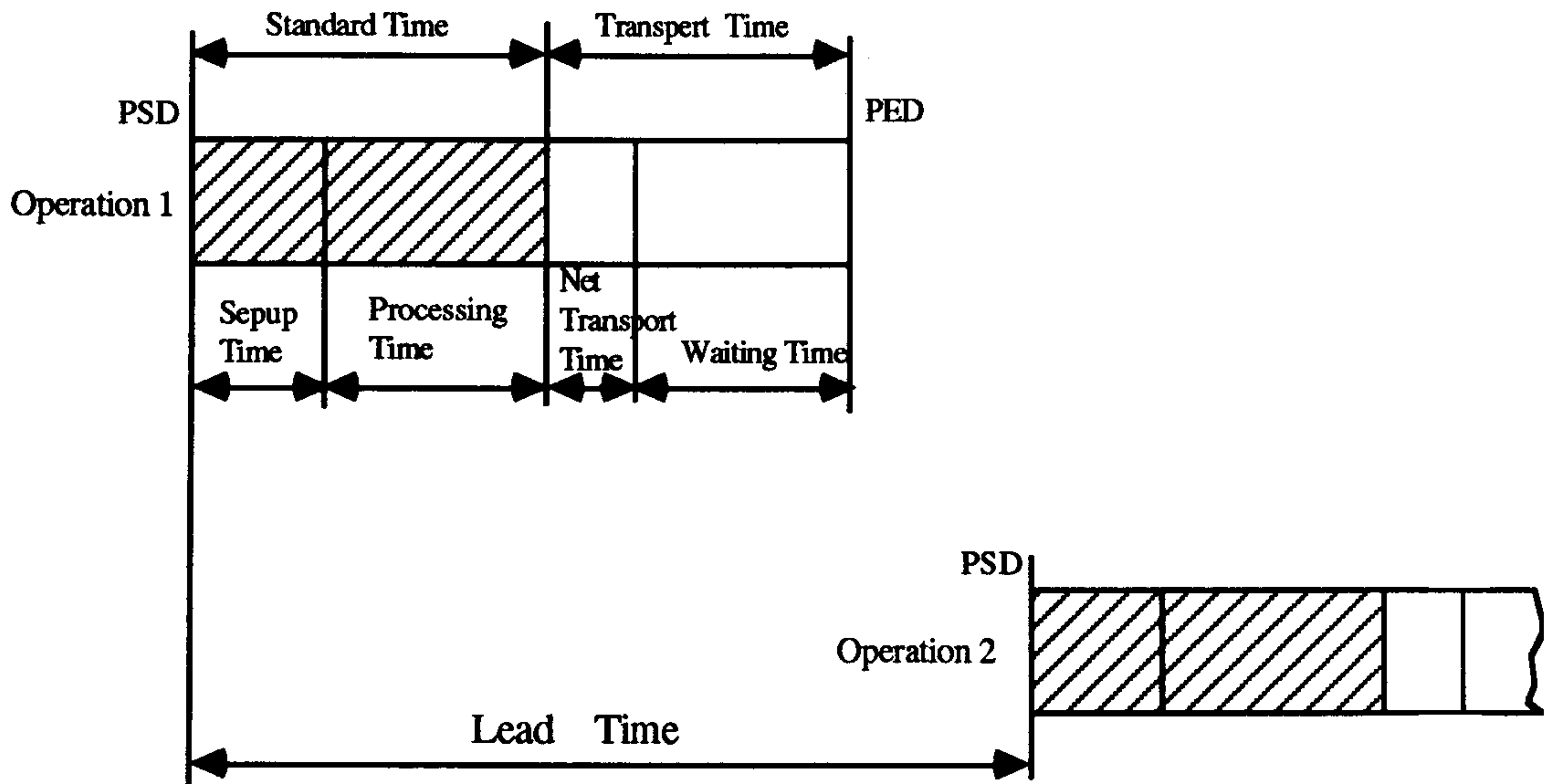
한 Machine에 대해 날짜별로의 부하가 수행해야 할 Operation에 대한 검증이 이루어진다. 검색하는 Logic의 차이일 뿐 기본적으로 앞서 설명한 7가지가 Check Point이다.

2. PERT(Program Evaluation and Review Technique)

Calculation

한 Order의 Planned Date가 주어졌을때 그 Operation들의 일정을 구하기 위해서 PERT 계산을 수행한다. 이때의 계산 방법에는 전방계획법(Forward Scheduling)과 후방계획법(Backward Scheduling)이 이용되는데 그 각각의 결과가 Earlist Date와 Latest Date이다. 이 Routine에서는 이러한 작업을 수행하기 위한 입력과 계산이 이루어진다.

가. Scheduling에서의 Time의 정의



<그림 4-4> Scheduling에서의 Time 정의

Scheduling시 Time은 <그림 4-4>와 같이 정의된다. 여기서 Standard Time의 정의는 다음과 같으며

$$\text{Standard Time} = \text{Setup Time} + \text{Processing Time}$$

그 값은 공정 관리자가 금형도와 Standard Time 산출표를 보고 결정한다. Transport Time은 순수 이송시간에 Waiting Time을 더한 것으로 이 Waiting Time은 현 공정의 기계가공이 끝난 직후

다음 공정의 시작 때까지 있어야 하는 최소한의 대기시간으로 역시 공정관리자가 산출한다. Lead Time은 공정과 공정간의 시간 간격으로 다음 식과 같다.

$$\text{Lead Time} = \text{Standard Time} + \text{Transport Time} + \alpha$$

여기서 α 는 Operation의 수행에 필요한 최소한의 시간 외에 부하 또는 다른 Order들의 Schedule을 고려한 추가 대기시간 (Waiting Time)이다.

1절에서 언급한 바 있는 Lead Time 계산은 엄밀히는 Standard Time과 Transport Time의 산출이며 실제 계산 관리되는 것도 이 값들이다.

나. PERT 계산을 위한 Data 입력

화면상에 Window를 띄우고 Table을 그린다. 그리고 여기에서 2장에서 설명한 바 있는 Keyboard 입력 Function을 이용하여 각 Field의 Data를 입력받는 기능이다.

여기서 입력되는 Data는 Setup Time, Processing Time, Transport Time, 그리고 Lead Time이다. 회사에 따라서는 Setup Time, Processing Time 대신에 Standard Time을 입력 받기도 한다.

Order의 Schedule 변경을 염두에 둘 때 Order에 대한 Lead Time도 입력의 대상이 된다. 이 경우 Mold와 Order의 관계는 Order와 Operation간의 관계가 되어 똑같은 과정을 적용할 수 있

다. 그러나 IPIS에서는 Mold의 Planned Date 결정은 경영계획으로 한번 결정된 것은 거의 Fix된 값으로 사용하며 Order의 Planned Date도 공정 관리자가 결정한 값을 거의 변경없이 쓴다고 생각한다.

PERT Data Input Screen

Order No : _____

Order PSD : _____ Order PED : _____

Unit : min

Operation No	Setup Time	Lot Size	Unit Processing Time	Transport Time	Lead Time
1					
2					
3					
4					
5					
6					

← → ↑ ↓ ↻

〈그림 4-5〉 Table Input의 화면 예

이 Function의 화면구성은 〈그림 4-5〉와 같다. 입력은 Order No와 그 Order의 Planned Date이다. 그리고 Sequential Number를 갖는 각 Operation에 대해 Setup Time, Transport Time 등을 입력한다. 입력시에는 Keyboard 입력 Function에서 설명한 Arrow

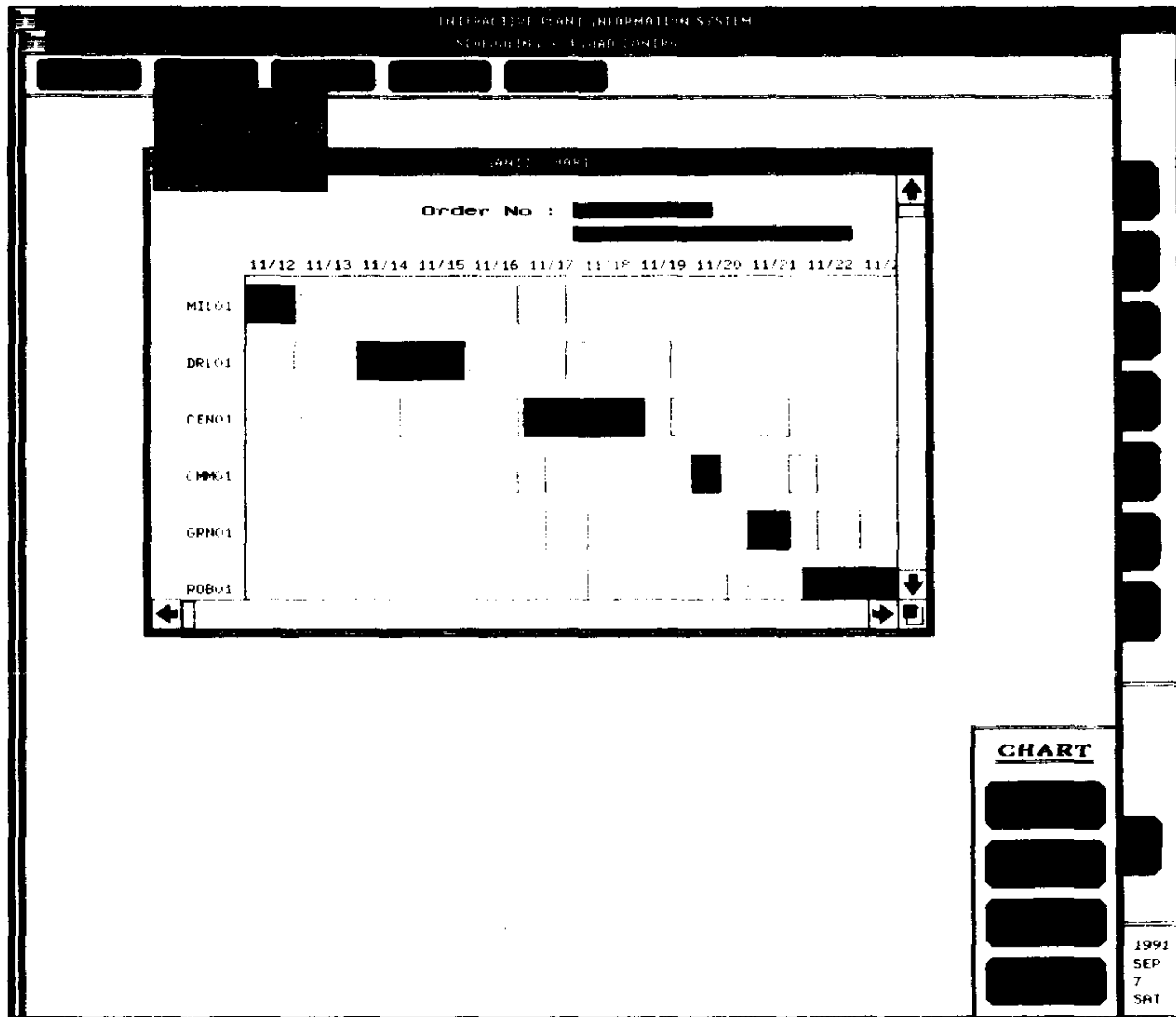
Key와 Control Key를 사용하여 일반 Screen Editing과 똑같이 수행한다.

Editing 기능의 완벽한 재현으로 File Input과 같은 Batch Processing 기능을 별도로 부여할 필요가 없어져 이 System에서는 그 기능은 개발하지 않기로 하였다.

다. PERT Calculation

필요한 Data의 입력이 끝나면 Operation들의 Planned Date를 계산하는데 Guide Line이 되는 Earlist Date와 Latest Date의 계산이 필요하다. 한 Operation의 수행에 꼭 필요한 시간은 Standard Time에 Transport Time을 더한 값이다. 이 값은 Operation Time이라고 정의한다.

Earlist Date의 계산은 Order의 PSD(Planned Start Date)부터 각 Operation 시간에 따라 배열하여 얻어진 시간이다. 그리고 Latest Date 계산은 Order의 PED(Planned End Date)로 부터 맨 나중 Operation 부터 Operation 시간에 따라 배열하여 얻은 값이다. 이것에 대한 화면이 <그림 4-6>에 나와 있다. 여기서 앞의 실선으로 된 Box는 Earlist Date의 시작과 끝을 나타내며 뒤의 실선으로 된 Box는 Planned Date의 시작과 끝을 나타낸다.



<그림 4-6> Earlist/Latest Date를 표시한 Gantt Chart

3. Automatic Scheduling

일정계획에는 작업을 완료할 공정순서 결정, 각 공정의 실행에 필요한 자원(기계, 공구, 치구 등)의 지정, 각 작업의 시작과 완료시간 지정 등이 포함된다. 이와 같은 일정 계획의 문제는 복잡하고 어려운 점이 많이 있는데 이를 정리하면 다음과 같다. 첫째, 일정 계획 문제의 가장 두드러진 특징은 문제의 폭발적 급증(Combinational Explosiveness)에 있다. 즉, 공정, 기계, 공구,

사람, 작업 등의 수에 따라 가능한 일정(Schedule)이 지수함수적으로 증가한다. 간단한 예로서 12 작업이 6 공정을 모두 거쳐 수행되는 개별 생산(Job Shop)의 경우 가능한 작업 배치의 방법은 $(12!)^6 (= 479,001,600^6)$ 이 된다. 그러나 같은 공정을 수행할 수 있는 대체 기계나 여러 다른 공정을 수행할 수 있는 기계가 있을 때에는 문제의 복잡성이 다소 완화될 수는 있다.

둘째, 생산에서 직면하는 일정 계획 문제는 새로운 작업의 우선 순위, 예기치 않은 변화에 대처하기 위해 현재의 작업 일정 계획을 수정해야 할 필요성이 대두된다. 이러한 재 일정계획을 위해 일정 계획기법은 관련된 모든 최초의 상태를 생산 시스템에 적용시킬 수 있는 능력이 있어야만 한다. 마찬가지로 생산일정은 일반적으로 한정된 생산기간에 대해 수립된다.

개별 생산에 있어서 주문의 수와 구성을 모를 경우 생산 기간은 가장 늦은 작업(Job)의 납기 또는 모든 작업의 생산주기(Makespan)로 정해진다. 개별 생산에 있어서 장기적으로 유향 상태에 있는 경우를 제외하고는 일정 계획기간 마지막 시점에서 유향 상태 또는 부하가 없는 경우는 없다. 따라서, 생산 시스템의 최종 단계에 적용 또는 지정할 수 있는 생산 일정 계획 기법이 필요하다.

셋째, 고전적인 기계 일정계획에서는 공정시간(Process Times), 작업 개시시간(Release Times), 납기(Due Dates)가 확정적(Deterministic)이며 작업, 기계, 인력, 자재 자원은 항상 이용 가능한 것으로 고려하나, 실제 생산환경은 공정시간, 작업개

시시간, 납기가 가변적일 뿐만 아니라 생산자원들이 이용 가능하지 않은 경우가 빈번히 발생한다. 따라서 생산 환경에서의 예기치 않은 사건과 변화를 나타낼 수 있는 일정 계획시스템이 요구된다.

Automatic Scheduling 기능의 가장 이상적인 경우는 이러한 역할을 충분히 수행할 수 있는 Scheduler가 되는 것이라 할 수 있다. 그러나 앞서도 언급한 바와 같이 이러한 기능을 갖추기 위해서는 많은 노력과 시간이 투자되어야 할 뿐만 아니라 여기에서 얻어진 값도 그 신뢰성이 떨어진다는 문제점이 있다. 이 때문에 IPIS에서는 Automatic Scheduling의 개발에 좀 더 신중을 기하기로 한다. 먼저 다음에 설명할 Interactive Scheduling과 Rescheduling 기능의 개발에 착수하여 그 결과를 보고 그때의 Sequence를 Algorithm화하여 Automatic Scheduling의 개발에 적용하기로 하였다.

4. Interactive Scheduling

앞서의 PERT Calculation을 통해 Order의 Planned Date와 Operation들의 PERT Data를 구했다. 이를 토대로 Operation의 Planned Date를 구하고자 하는 것이 여기서의 기능으로 그 값의 최적치를 구한다기 보다는 적당한 선에서의 Assign 과정이라 할 수 있다.

가. Criterion Input

Planned Date 결정시 Forward Scheduling을 따를 것인지 아니면 Backward Scheduling을 따를 것인지에 대한 Criterion을 입력 받는다. Forward Scheduling은 Order의 Planned Start Date로부터 Operation Sequence에 따라 Planned Date를 결정해 나아간다. 이 과정에서 Assign할 Machine의 부하 상태에 따라 원하는 시간에 수행을 할 수 없는 경우에는 그만큼 일정을 뒤로 Shift한다. 이러한 과정을 반복하여 맨 마지막 Operation의 Planned End Date가 Order의 PED를 넘기지 않은 경우에 이 Scheduling은 완료된다. 반면, Backward Scheduling은 Order의 Planned End Date로부터 Operation Sequence의 역순으로 Planned Date를 결정하는 방법이다. 마찬가지로 부하 상태등으로 인해 원하는 시간에 작업이 불가능한 경우에는 앞으로 그만큼 Shift시키며 맨 처음 Operation의 Planned Start Date가 Order의 PSD를 넘기지 않은 경우에 이 Scheduling은 완료된다.

Forward Scheduling을 하는 경우 납기 전에 미리 만들어 둬으로써 납기보다 여유있는 작업이 될 수 있으나 대량 생산인 경우 재고 문제가 발생하여 창고 비용 등으로 인한 손실을 초래한다. Backward Scheduling의 경우 납기에 꼭 맞추어 제작하므로 재고 문제는 없으나 돌발 사고 등으로 납기 지연이 발생한 경우는 문제를 복잡하게 한다. 금형 공장의 경우 대부분의 Lot Size는 1개로, 이로 인한 재고 문제는 심각하지 않다고 보고 Forward

Scheduling을 주로 택한다. 따라서 Default도 Forward Scheduling으로 잡았다.

나. Scheduling

PERT Data Input 과정에서 입력된 Order들에 대해서 Planned Date를 결정한다. 수행 방법은 Order별로 수행하는데 이때의 순서는 입력순이거나 Order의 Priority가 높은 순으로 수행한다. 여기서의 Priority는 Order 입력시 User가 적용한 것이다.

한 Order에 대한 Scheduling의 단계는 다음 순서와 같다.

- Order의 Scheduling Criterion으로 Forward Scheduling을 택한다.
- Forward Scheduling의 경우 가장 이상적인 Planned Date는 PERT Calculation에서의 Earliest Date와 같은 값을 갖는다.
- 주어진 Order의 첫번째 Operation은 그것의 Earliest Date가 Planned Date가 되고 수행해야 할 Machine은 Parallel Machine Group내의 첫번째 기계이다.
- 그 첫번째 기계의 Load Profile을 화면에 띄워 원하는 날짜에 Operation의 수행이 가능한지를 살펴보고 가능하면 현재의 Date를 Planned Date로 삼고 그렇지 않으면 같은 Group내의 다음 기계에 대한 Load Profile을 검사한다.
- Group내의 모든 기계에서 앞서 지정한 날짜에 수행하는 것이 불가능하면 이 Group내의 기계들 중 제일 먼저 수행

가능한 상태가 되는 기계와 그 시간을 구한다.

- 첫번째 Operation에 여기서 구해진 Machine이 Assign되고 그 시간이 Planned Date로 등록된다.
- 두번째 Operation의 Planned Date는 첫번째 Operation이 끝나는 시간이 Start Date가 되고 여기에 Operation Time을 더한 것이 End Date가 된다.
- 이 Planned Date를 가지고 수행 가능한 기계가 있는지를 앞의 방향으로 조사하고 없으면 가장 먼저 나타나는 수행 가능 시간을 Planned Date로 결정한다.
- 이 방법을 반복하여 맨 마지막 Operation의 Planned End Date가 Order의 Planned End Date를 Over하지 않았는지 Check한다.
- Over하지 않았으면 이 Order의 Scheduling은 종료되며 그렇지 않은 경우에는 각 Machine의 부하 용량을 표준치 이상(120 - 150%)으로 가정하여 다시 한번 위의 과정을 반복한다.

이상의 방법을 거쳐 Scheduling이 완료되는데 유의할 점은 기존 부하나 다른 Order의 Schedule에 대한 변경은 전혀 고려하지 않는다는 점이다. PERT Data Input 과정에서 입력된 모든 Order에 대한 위의 작업이 끝나면 이 Routine의 수행도 끝나게 된다.

Interactive Scheduling의 결과는 부하 조정을 거쳐 최적화를 피하기 보다는 각 기계에 과부하를 허용하는 편법을 써서 얻어진 중간 결과이다. 따라서 이 결과는 차후의 수행을 위한 Data라기

보다는 작업의 배열과 이를 통한 현장 상황의 Update라는 의미를 지니며 이에 대한 수정을 필요로 하는데 이 작업은 다음의 Rescheduling Routine에서 수행할 것이다.

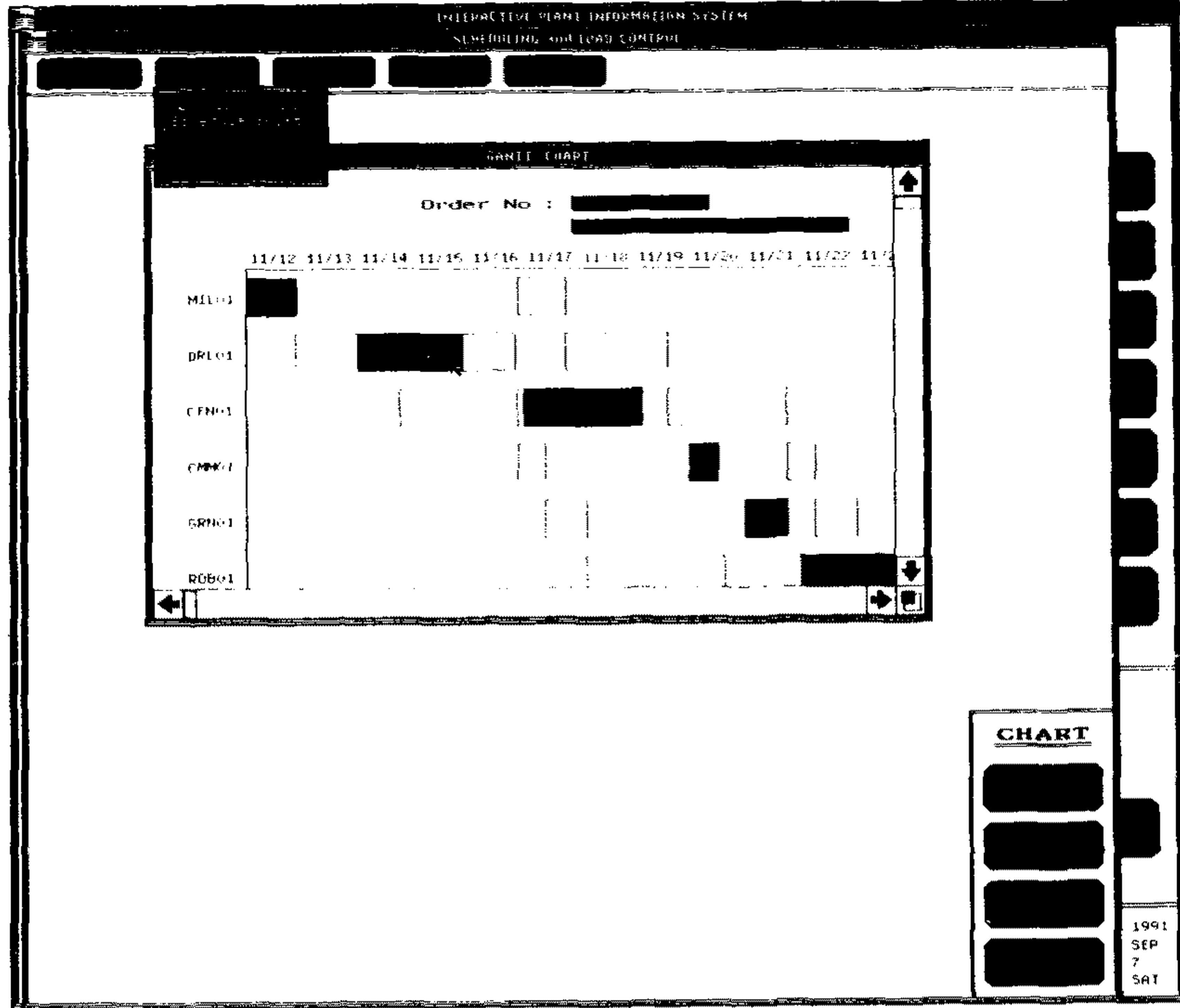
5. Rescheduling

Interactive Scheduling은 각 Operation의 배열만을 염두에 둔 대강의 Scheduling이었다. 이제 이를 재 계획하여 실제 수행 가능한 Schedule로 만드는 작업이 필요한데 이 Routine에서 이 작업이 수행된다.

가. Schedule Shift

Schedule Shift의 수행은 Order별로 한다. 해당 Order의 Gantt Chart를 <그림 4-7>과 같은 화면에 띄우고 Menu를 이용하여 시간축상에서 좌우로 움직인다. VAXstation Screen 상단의 Menu를 따라 순차적으로 진행하면 Shift Menu를 만나게 된다. 이것을 Click하면 Gantt Chart내의 Planned Date를 나타내는 Box가 Enable되어 Mouse의 움직임을 감지한다. 원하는 Planned Date의 Box에서 Mouse의 맨 왼쪽 Button을 누르고 누른 상태에서 그 Box를 원하는 날짜, 원하는 시간만큼 좌우로 Shift한다. Shift가 수행되어지는 경우 전후 Operation에 대한 Schedule Propagation이 일어나야 하는데 이에 대한 Algorithm은 다음에 설명하기로 한

다. 이상의 수행은 Memory Allocation을 통해 이루어지는데 Speed 향상 및 Undo 기능 등의 수행을 쉽게 한다.

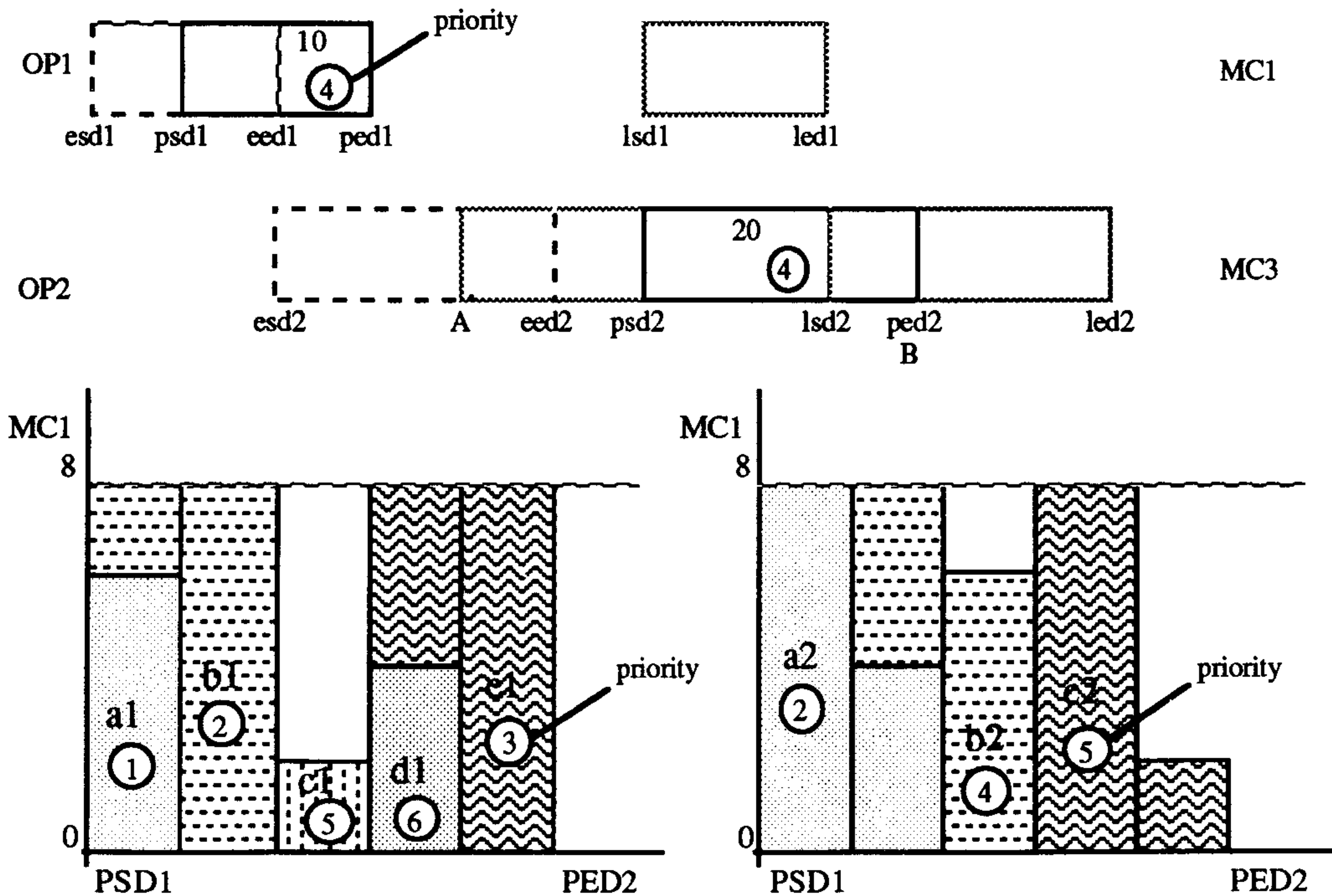


<그림 4-7> Schedule Shift의 화면 예

(1) Shift Propagation

아래의 <그림 4-8>은 Schedule Propagation을 설명하기 위한 예제이다. OP1, OP2는 주어진 Order의 Operation 1과 Operation 2이며, MC1, MC2는 그들을 수행하기 위한 Machine 1과 Machine 2이다. 성긴 점선으로 그려진 Box는 Earlist Date로 왼쪽변은 Earlist Start Date, 오른쪽변은 Earlist End Date를 나타낸다. 촘촘한 점선으로 나타낸 Box는 Latest Date를 나타내는 것으로

역시 왼쪽변은 Latest Start Date, 오른쪽변은 Latest End Date를 나타낸다. 그리고 실선으로 나타낸 것은 Planned Date로 왼쪽변은 Planned Start Date, 오른쪽변은 Planned End Date를 나타낸다. OP1, OP2의 Planned Date안에 들어있는 숫자는 Operation Time을 의미하며 원 속에 들어 있는 숫자는 그들의 Priority이다. Machine의 Load Profile에서도 원속의 숫자는 Priority이고 시간축의 PSD1, PSD2는 각각 OP1, OP2의 Planned Start Date이다. 그리고 Load Profile에서 8은 정규 시간을 나타내며 이들은 점선으로 나타내었다.



<그림 4-8> Schedule Propagation의 예

OP1의 Schedule을 Shift한다고 할때 이 작업은 두 단계로 나누어 생각할 수 있다.

A, B가 아래에서처럼 정의되고 PSD1을 기준으로 생각할때

$$A = PSD2 - (PED1 - PSD1)$$

$$B = LED2 - (PED1 - PSD1)$$

1) 다른 Operation에 영향을 주지 않는 Schedule Shift :

ESD1 - A 사이의 Shift

2) 다른 Operation에 영향을 주는 Schedule Shift :

A - B 사이의 Shift

의 두가지로 나누어진다. 이 두 경우 모두 Load Profile상에서 적당한 수행 가능 위치를 찾을 수 없을때에는 여기에 있는 다른 Order의 Operation의 Schedule을 조정해야 하는데 Schedule Shift의 세번째 단계 이것이다.

3) 다른 Order의 Operation Schedule Change

Load Profile을 조회해 가면서 수행 가능한 기계를 찾는 경우가 다음에 설명할 Machine Allocation의 기능이다. 앞의 세가지 기능을 수행하다 보면 Schedule Shift와 Machine Allocation이 독자적으로 수행할 수 있는 것이 아니라 동시에 수행되는 기능임을 알 수 있다. 따라서 이 내용을 여기서 함께 설명하기로 한다.

(가) 다른 Operation에 영향을 주지 않는 Schedule Shift

수행하고자 하는 Operation이 전·후에 있는 Operation들과 일정의 Overlap이 발생함이 없이 재 계획되어지는 구간이다. 따라

서 해당 Operation의 Schedule만 고려하면 되므로 문제가 단순하고 수행이 빠르다. 이 과정의 수행 단계는 다음과 같다.

- 지정된 Machine MC1의 Load Profile을 보면서 10시간의 부하를 갖는 OP1이 들어갈 자리를 찾는다.
- MC1의 Load Profile에서 PSD1 날짜 시작위치에 부하가 있는지를 살펴 본다.
- 부하가 없으면 10시간의 부하가 들어갈 수 있는 자리인지를 살펴 보고 그렇지 않으면 Load Profile상의 다음 부하가 끝나는 위치를 조사한다.
- MC1의 Load Profile상에서 a1, b1, c1, ... 순으로 그 끝나는 시간에 Free Time이 있는지를 조사하면서 A 날짜까지 계속한다.
- 이상의 MC1에서 적당한 자리를 찾지 못하면 그 Parallel Machine인 MC2에 대해서 위의 과정을 반복한다. -

Alternative Machine에서의 수행 내용

- 모든 Parallel Machine에 대해서 수행해도 적절한 위치를 찾지 못하는 경우 가장 긴 Free Time이 있는 Machine에서 얼마간의 Overtime을 허용하여 수행해 낼 수 있는지를 Check한다. - Overtime의 수행내용. 이 단계에서는 수행되지 않는 내용임.

(나) 다른 Operation에 영향을 주는 Schedule Shift

수행하고자 하는 Operation이 다음 Operation과 Overlap되는

경우 그 Priority에 따라 현재 Operation의 Shift를 허용 불가하든지 다음 Operation의 Schedule을 뒤로 그만큼 Shift시키든지 하여야 한다. 이에 대한 수행 방법은 다음 Step과 같다.

- 수행 Operation과 다음 Operation의 Priority를 비교하여 수행할 Operation의 Priority가 다음것보다 같거나 크면 다음을 수행한다.
- MC1 또는 Parallel Machine Group내의 다른 기계에서 A 날짜를 지나 B날짜까지 각 부하가 끝나는 자리를 조사 하였더니 Free Time이 10시간 이상인 위치를 발견하였다.
- 이 위치에 OP1의 Schedule을 정하고 OP2의 Shift를 고려 한다.
- OP2의 다음 Operation은 OP3라 하고 OP2의 수행 Machine 은 MC3이며 그 Parallel Machine으로 MC4가 있다고 할때 (가)에서와 같은 Analogy가 형성된다.
- OP2에 대해서 (가)의 Logic을 수행해보고 적당한 위치가 결정되면 그 자리가 OP2의 Schedule이 되고 그렇지 않으면 앞서 결정된 OP1의 Schedule은 취소되고 다른 기법을 동원하여 OP1의 일정을 구함

(다) 다른 Order의 Operation Schedule Change

Load Profile을 살펴보면 Free Time이 곳곳에 자투리로 남아 있는 경우가 있어 부하들 중 일부의 일정을 조정하면 원하는 시간 만큼의 Free Time을 얻어낼 수 있는 때가 있다. 이 작업을 수

행하는 것이 여기서의 작업이며 다음 Step을 거친다.

- MC1의 PSD1 부터 주어진 Operation OP1의 Priority보다 낮은 값을 갖는 Order의 Operation을 찾는다.
- a1, b1은 OP1의 Priority보다 높고 c1의 Priority는 더 낮다는 것이 조사되었다.
- c1을 Load Profile에서 삭제하면 Free Time이 6시간에서 8시간으로 늘어난다. 이 값은 OP1의 Operation Time 10시간을 충족시키지 못하므로 다음 부하를 조사한다.
- d1의 Priority 역시 OP1보다 낮으므로 이것을 Profile에서 삭제하고 Free Time을 조사한다. 이 경우 Free Time은 12시간으로 늘어났다.
- Free Time이 충분하면 여기에 OP1의 일정을 위치시키고 그렇지 않으면 A의 날짜까지 수행한다.
- MC1에서 원하는 결과를 얻지 못하면 Parallel Machine인 MC2에 대해 위의 과정을 반복한다.
- 모든 Parallel Machine에 대한 조사에서도 원하는 결과를 얻지 못하면 Shift 범위를 늘려 A날짜에서 B날짜 사이의 구간에 대해서도 위의 과정을 반복한다.
- OP1의 일정이 결정된 경우, 이를 수행하기 위해 Profile에서 지워졌던 부하들에 대해서 재 계획이 이루어져야 한다. 이 경우에도 Priority에 따라, 또는 작업 시간이 짧은 순서에 따라 수행하는데 여기서는 c1이 먼저 고려의 대상이 된다.

- c1의 수행은 (가)에서 설명한 Logic만을 수행하며 (나)나 (다)까지는 고려에서 제외한다. 왜냐하면 이러한 경우, 그 경우의 수가 기하급수적으로 늘어나 처리 불능 상태에 빠지기 때문이다.
- c1의 재 계획이 안되는 경우에는 OP1의 Shift를 취소하거나 Rescheduling의 다른 기법을 동원한 의사 결정이 필요하다.
- d1에 대해서도 c1과 같은 수행을 한다.

(2) Schedule Shift를 위한 보조 기능

Schedule Shift를 수행하기 위해서는 Order의 Gantt Chart나 Machine의 Load Profile에 대한 화면 출력과 같은 기능 외에 몇 가지 보조 기능을 생각해 볼 수 있다.

1) 주어진 Operation의 Shift 가능범위에서 Parallel Machine Group내의 총부하를 구한다. 이를 토대로 원하는 작업 시간만큼의 무부하 시간이 있는지를 조사한다. 따라서 주어진 시간 간격내의 부하율을 구하는 기능이 필요하다.

2) 모든 Operation에 Priority를 부여할때 무부하시간(Free Time)은 가장 낮은 Priority를 갖는다. 이 개념으로부터 주어진 Priority이하로 구성된 연속된 부하들의 조합을 구하고 이 조합들이 갖는 전체 시간을 구해 주는 기능이 필요하다. 즉, 이 조합중에서 어느 하나에 현재의 Operation을 Assign한다.

3) Overtime을 부여하는 경우 Free Time은 그만큼 늘어난다.

어떤 시간 간격에 걸쳐 Overtime을 주는 경우 무부하시간을 갖고 끝나는 경우는 Free Time은 Overtime을 더한 것만큼 늘어난다. 이를 이상에 걸리는 부하가 있는 경우는 부하의 조정을 거친 후 Free Time이 구해진다. 따라서 Overtime이 주어진 경우 Free Time을 다시 구하는 기능이 필요하다.

나. Alternative Machine

공작물에 구멍을 뚫는 작업이라면 먼저 떠오르는 것이 Drilling Machine일 것이다. 그러나, Drilling Machine이 없는 경우에는 Milling Machine, Machining Center 등에서도 같은 작업을 수행할 수 있다. 경우에 따라서는 선반에서도 구멍뚫기 작업이 가능하므로 Drilling Machine의 대체 기계는 여러가지라고 생각할 수 있다.

특정 Operation을 수행하기 위해 지정된 Machine에 갔을 때 이 기계의 부하가 많아 대체 기계에서라도 작업을 수행하고자 하는 경우가 있을 것이다. 이때 대체 작업이 가능한 기계를 Alternative Machine이라고 부른다. 이러한 Alternative Machine은 실제로 작업은 가능하지만 이들을 모두 관리하기는 어렵다. 그래서 Size별로, Type별로 호칭은 다르지만 같은 계열의 Machine들을 Group으로 묶어 관리하는 경우가 많은데 이 Group을 Parallel Machine Group이라 하며 Scheduling시 대체 기계는 이 Group에서 찾도록 한다.

대체 기계를 찾는 방법은 가.의 Schedule Shift에서 충실히 설명하였다. 부연하면 지정된 기계에서 원하는 기간내에 적당한 위치를 찾을 수 없으면 Parallel Machine Group내의 다음 기계를 찾아 간다. 그리고 앞의 수행을 반복하여 현재의 Operation을 찾는 작업이 이 Routine에서의 기능이다. 이때 Speed 향상을 위해 Parallel Machine들에 대한 Data를 Memory Allocation 기능을 사용하여 Main Memory에 저장시켜 놓고 수행하기로 한다.

해당 기계가 찾아진 경우에는 그 기계에 Operation을 부하로 등록한다. 이것을 Operation 관점에서 보았을때 Machine Allocation작업이라 한다.

다. Transport Time Change

Transport Time의 정의는 <그림 4-4>에 나와 있다. 이것은 공정관리자에 의해 결정되는 공정간에 주어져야 하는 최소한의 대기 시간이라고 할 수 있다. 하지만 그 Order가 Rush Order로 바뀌는 경우등에는 최소한의 개념이 아닌 더욱 짧은 시간이 되어 극단적으로는 순수 이송 시간만 남게 되는 경우가 될 것이다. 이처럼 Order 또는 Operation의 긴급도에 따라 Transport Time을 변경할 경우가 생기는데 보통 줄이는 방향으로 수행된다.

Transport Time의 Change는 Mouse의 Click이나 직접적인 Keyboard 입력으로 이루어지는데 이 결과로 Operation Time이 변화한다. Operation Time이 줄어들면 상대적으로 Shift할 영역이

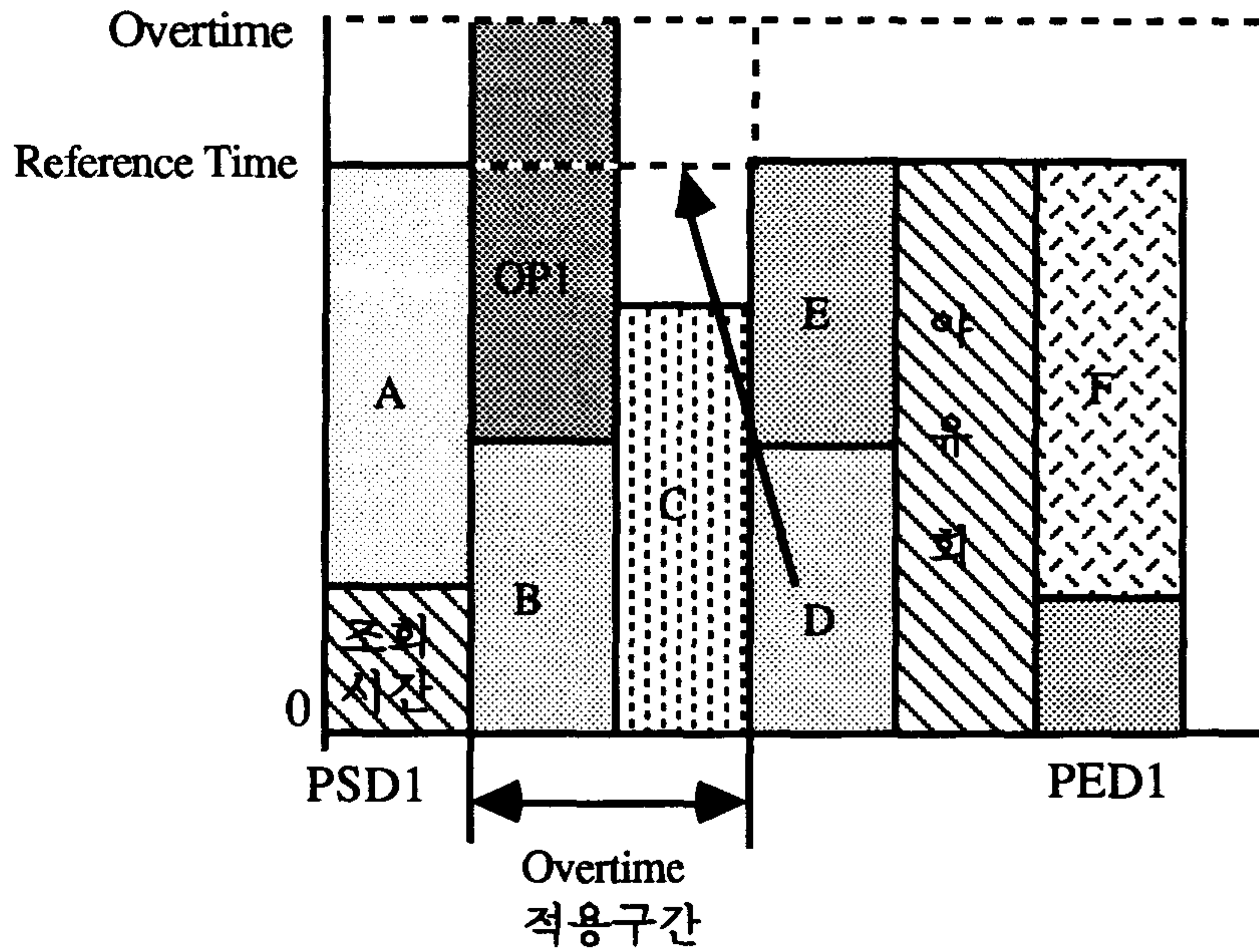
넓어지고 보다 작은 Free Time에서도 일정을 결정할 수 있다는 장점이 있다.

Transport Time을 변경하고 다음의 Scheduling 작업은 가.에서 설명한 Schedule Shift의 방법을 그대로 적용한다. 유의해야 할 사항으로 Transport Time을 변경한 경우 단순히 해당 Operation에만 영향을 주는것이 아니라 그 Order의 Earliest Date와 Latest Date의 변경을 가져와 전체적으로 Scheduling을 용이하게 해준다.

라. Overtime

정규시간을 갖고 작업하는 경우 도저히 작업의 소화가 되지 않을때 Overtime을 적용한다. Overtime의 적용은 현장 전체에 대해 하는 경우와 병목 현상을 보이고 있는 작업장에 대해서만 한정적으로 실시하는 경우가 있다. 이 두가지 경우에 대한 Database상의 처리는 Factory Calendar를 이용하여 수행될 것이다.

Overtime을 적용하는 경우 Machine의 Load Profile에서 적용 구간만큼 Queue가 커진 것으로 생각할 수 있다. 따라서 이의 적용을 위한 특별한 Logic은 필요하지 않다. 다음 <그림 4-9>는 Overtime의 적용 예를 보인 것으로 정규시간만으로는 처리 불가능했던 OP1의 일정이 쉽게 결정되고 D의 작업을 하루 앞당겨 실시할 수 있다는 내용을 보여 주고 있다.



<그림 4-9> Overtime의 적용 예

마. Priority Change

현장에서 흘러가고 있는 많은 Order들은 그 긴급도에 따라 Priority를 갖는다. 뿐만 아니라 같은 Order내의 Operation들이라 할지라도 작업의 지연, 지정된 기계의 부하에 따라 다른 Priority를 갖게 된다. 이러한 사항을 수치적으로 표현하기 위해서는 하나의 식이 필요한데 IPIS에서는 다음과 같은 식을 제안한다.

$$\text{Priority} = \left\{ \frac{(\text{Order PED} - \text{총 Operation Time})}{(\text{Order PED} - \text{Order PSD})} \right\} \times 100$$

즉, Order의 여유 시간을 무차원화하여 0 - 100까지의 수치로 바꾼 것이다. 이 외에도 여러가지의 Priority Rule이 있는데 여기서는 이것만 고려하기로 한다.

Priority가 Scheduling에서 주는 의미는 가.의 Schedule Shift에서 충실히 설명하였으므로 여기서는 생략하기로 한다. 이러한 Priority는 공정관리자에 의해서 조정되고 그 영향은 차후의 Scheduling에 계속 영향을 미친다. 한 예로 결정된 Schedule에 대해 변경을 불허하고 싶은 경우에는 Priority를 높여주면 된다. 그리고 긴급 Order인 경우에 높은 Priority를 갖고 Schedule에 임하면 우선적으로 각 Machine에 Assign된다.

바. Due Date Change

현재의 납기로는 도저히 생산이 불가능한 경우 Due Date Change를 수행한다. 이 경우 주문을 한 고객이나 회사의 경영계획과 상충되는 일이 발생할 수 있으므로 그들과의 조정이 필요하다. 따라서 이 과정은 Scheduling의 거의 마지막 수순에 해당한다.

Due Date의 변경은 주로 뒤로 미루는 작업이며 Keyboard 입력을 통해 수행된다. 납기가 늘어난 경우 Latest Date에 대한 계산이 다시 이루어져야 하며 그만큼 Operation들의 Schedule Shift

영역이 늘어나 재 계획을 용이하게 해준다.

사. External Contract

현장의 부하가 팽차서 더 이상의 작업이 어려운 경우나 현장에 필요한 장비가 없어 수행이 불가능한 Operation에 대해서 외주를 주도록 한다.

Order에 대해서 외주를 주는 경우에는 예하의 모든 Operation Record들을 삭제하고 Order의 외주 Code를 Set한다. Operation의 외주인 경우는 해당 Operation Record의 외주 Code를 Set하고 PSD를 외주 발주일, PED를 납기로 등록한다. 그리고 Machine에 관계되는 Field는 Erase한다. 여기서 PSD, PED의 조작을 통한 Gantt Chart에서의 Shift는 허용되나 Load Profile에 대한 고려는 필요없다.

외주 역시 그 결정은 상급 계획담당자가 수행한다.

6. Dispatch List Release

현장의 작업 반장 또는 Machine Operator에게 그날그날의 작업을 할당해주는 것이 작업지시서이다. IPIS에서는 일상적으로 보내는 Standard Form과 특별히 Comment를 줄수 있는 Additional Remark Form의 두가지를 만들었다. 그 각각의 내용은 다음과 같다.

가. Standard Form

- Component No, Order No, Operation No
- 작업 내용 - 80자 안팎으로 수행 내용 서술
- Standard Time, 작업 착수 시간, 작업 종료 시간,
Priority
- 공구 List, Fixture List, NC Program No
- 부서 번호, 작업장 번호, Machine No, Operation No
- Alternative Machine No - 고장 등 비상시 제일 먼저 수행
가능한 Machine No

나. Additional Remark Form

- 작업 수행에 관한 추가적인 내용을 160자 안팎으로 서술
 - 구체적인 Operation 방법 등 명시
- 작업의 긴급도에 관한 추가적인 내용 서술
- Machine 상태 등 현장에 관한 추가적인 내용 서술
- 기타 작업자에게 하고자하는 추가적인 서술

제 5 장 NC 프로그램 관리 및 분배

제 1 절 NC 프로그램 관리 및 분배 모듈의 역할

NC 프로그램 관리 및 분배 모듈의 기본 목적은 가공정보(NC Program)를 관리하고 생산 공정으로 이를 분배하며, 생산 공정으로부터 현황 정보를 실시간(Realtime)으로 수집하여 IPIS Database에 저장함으로써 공정 관리실과 현장(Process)과의 On-Line Link를 형성하도록 하는 것이다.

플랜트 통제 시스템(IPIS System)내에서 NC 프로그램 관리 및 분배 모듈이 수행하는 기능들을 금형의 수주에서 생산 공정까지의 측면에서 살펴보면 IPIS의 작업 관리 모듈을 통해 수주받은 금형의 각 부품에 대한 Order와 Operation들이 입력됨과 아울러 대략적인 작업 개시일과 종료일이 결정되고 Machine의 할당이 이루어진다. 작업 관리 및 제어 모듈에서 생성된 공정 정보(작업 개시일, 종료일, 공정이 수행될 Machine)를 통해 NC 프로그램 관리 및 분배 모듈에서는 Schedule File를 생성하여 해당 작업일에 각 공정으로 가공 정보(NC Program)를 On-Line으로 전송하며, 이러한 NC 프로그램들은 CAD/CAM 장비를 통해 생성되어지며 이 모듈에서 저장 관리 된다. 또 현장의 각 공정으로부터 실시간으로 전송되어오는 현황 정보를수집하여 IPIS Database에 저장함과 아울러 그래픽 또는 텍스트 출력을 통한 공정의 상태를 관리

(Monitoring)할 수 있게하며, 실적 관리 모듈을 통한 통계 데이터의 집계 및 관리와 작업 관리 및 제어 모듈을 통한 Schedule 변경등에 정보를 제공한다.

NC 프로그램 관리 및 분배 모듈에서는 DNC(Distributed Numerical Control)의 실현을 통해 다수의 NC 공작기계들에 NC 프로그램을 On-Line으로 전송하고 공정의 진행 상태를 관리할 수 있는 환경을 조성 하도록 하였다.

제 2 절 NC 프로그램 관리 및 분배 모듈의 기능 및 구성

1. NC 프로그램 관리 및 분배 모듈의 개발 환경

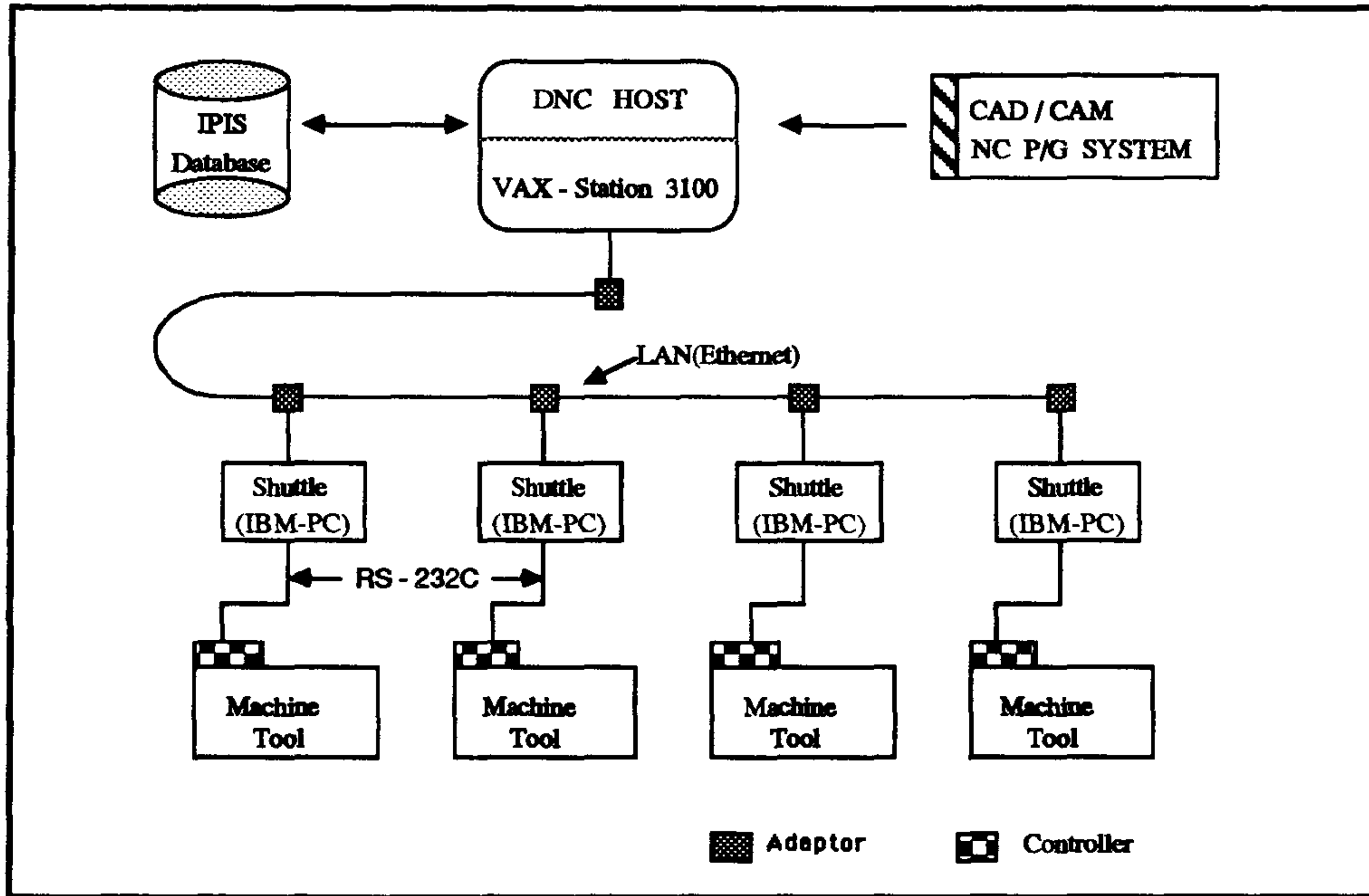
가. 하드웨어 구성(Hardware Configuration)

NC 프로그램 관리 및 분배 모듈의 기능을 수행하기 위한 주요 하드웨어 구성 요소는

- 중앙 컴퓨터(VAX-STATION 3100)
- 위성 컴퓨터(IBM-PC)
- 공작기계(CNC, ROBOT, CMM)
- LAN(Ethernet)/MAP, RS-232C Cable

등이며, 이들의 구성 및 규모는 IPIS System이 수행해야할 기능에 따라 달라질것이다. IPIS Model Plant는 다음 <그림 5-1>과 같으며, 중앙 컴퓨터와 위성 컴퓨터간의 통신은 LAN(Ethernet)

또는 MAP에 의해 연결되고, 위성 컴퓨터와 단위 공작기계간의 통신은 RS-232C로 연결하여 수행한다.



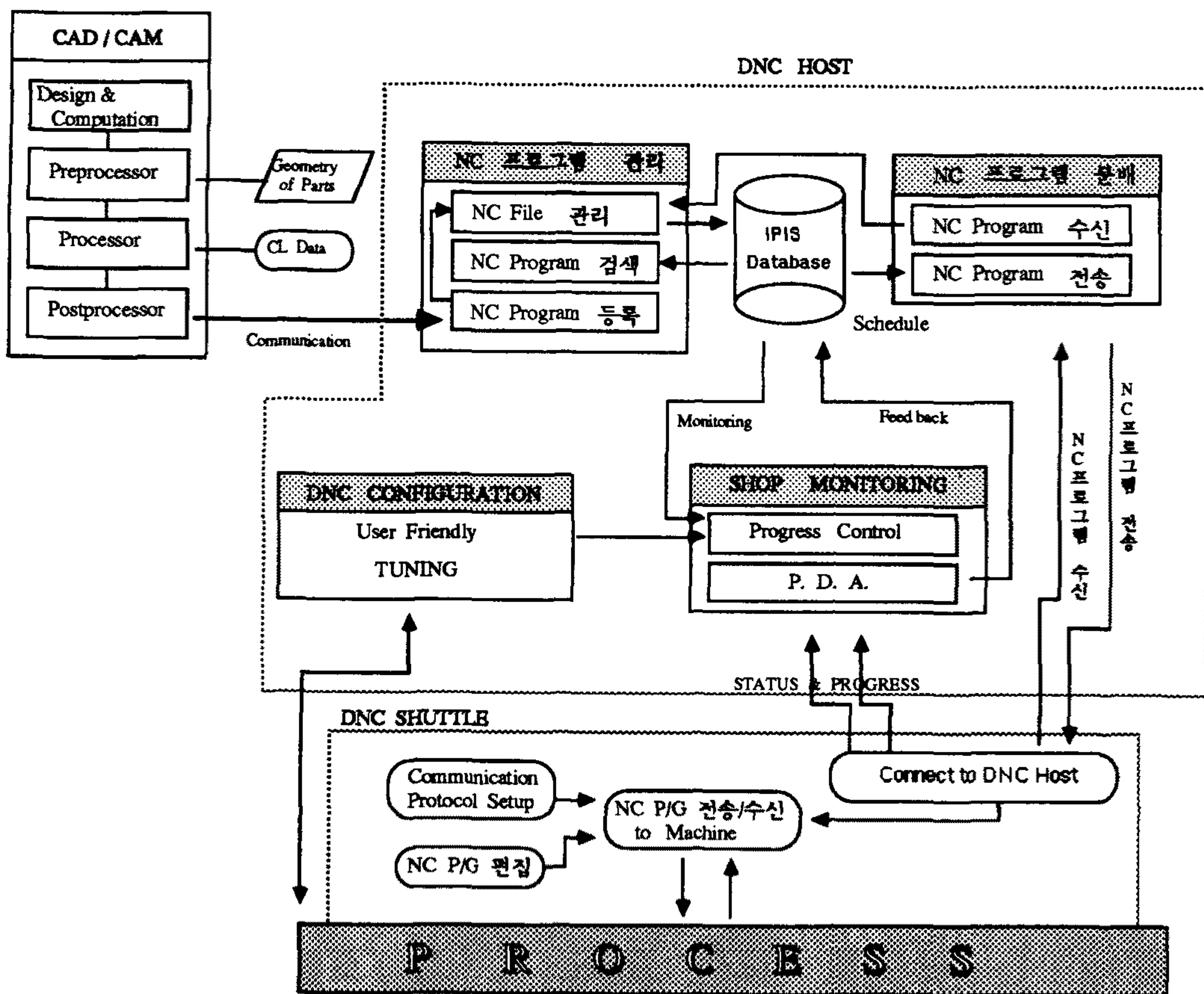
<그림 5-1> 하드웨어 구성도 (Hardware Configuration)

나. 소프트웨어 구성 (Software Configuration)

본 NC 프로그램 관리 및 분배 모듈은 중앙 컴퓨터 (Host Computer)에서 운용되는 기능과 위성 컴퓨터 (Shuttle)에서 운용되는 기능에 따라 각 컴퓨터에서 Application Program를 작성하였으며, 중앙 컴퓨터에서는 VAX-C 와 Rdb 그리고 Graphic 지원을 위한 VWS를 사용하고, 실시간 제어를 위하여 멀티-태스크 (Multi-Task)로 구성 하였으며, 위성 컴퓨터에서는 Turbo-C를 이

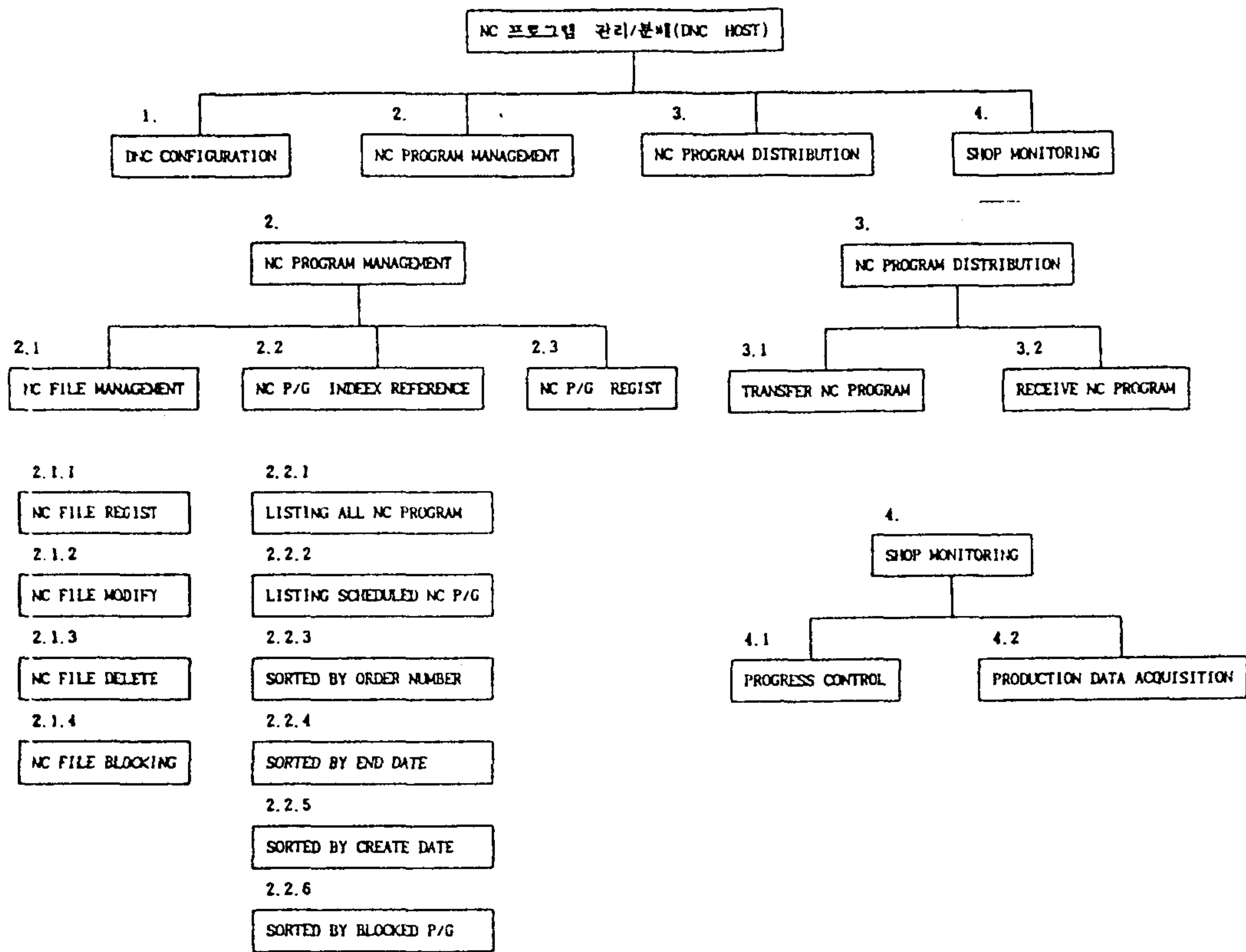
용하여 메뉴(Menu) 구성 및 위성 컴퓨터의 기능들을 수행하였다.

본 모듈에서 개발한 소프트웨어의 구성은 다음 <그림 5-2>와 같다.



<그림 5-2> 소프트웨어 구성도(Software Configuration)

2. 중앙 컴퓨터의 기능 구성

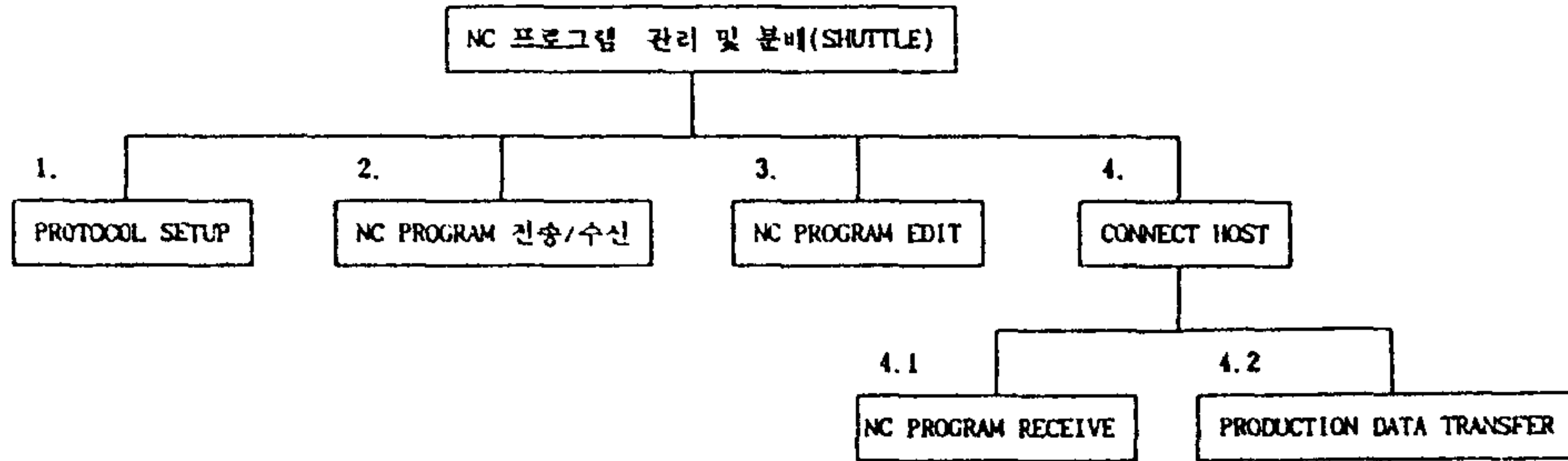


<그림 5-3> NC 프로그램 관리 및 분배 모듈의 중앙 컴퓨터 VTOC

〈표 5-1〉 중앙 컴퓨터의 기능 설명

Module		기능 설명
DNC Configuration 모듈		현장에 설치되어져 있는 각 기기들의 layout를 심볼화된 아이콘(ICON)를 이용하여 사용자가 대화식으로 Graphic 화면 구성
NC 프로그램 관리 모듈	NC File Menagement	NC 프로그램 관리를 위한 데이터의 입력, 수정, 삭제, 정지(Blocking)
	NC Program Index Referance	NC 프로그램의 조회 출력(Display)
	NC Program Regist	CAD/CAM 장비나 NC Programming 장치로부터 NC 프로그램의 수령
NC 프로그램 분배 모듈	NC Program Transfer	스케줄 (전송 예정 날짜/시간)에 따라 해당 위성 컴퓨터로 NC program 전송
	NC Program Receive	해당 위성 컴퓨터로부터 NC program 수신
현황 관리 모듈	Production Data Acquisition (P.D.A)	실 시간으로 위성 컴퓨터로부터 전송되어 오는 현황 정보의 수신
	Shop Monitoring	현황 정보를 텍스트(text) 또는 그 그래픽(graphic)으로 출력.

3. 위성 컴퓨터의 기능 구성



<그림 5-4> NC 프로그램 관리 및 분배 모듈의 위성 컴퓨터 VTOC

<표 5-2> 위성 컴퓨터의 기능 설명

Module	기능 설명
Communication Protocol Setup	위성 컴퓨터에 접속된 공작기계와의 통신 프로토콜을 설정(setting)
NC 프로그램 전송 / 수신	위성 컴퓨터에 접속된 공작기계로 NC 프로그램을 전송 또는 수신
NC 프로그램 수정	NC 프로그램을 일부 수정하거나 간단한 NC 프로그램의 작성
중앙 컴퓨터와의 접속(Connection)	중앙 컴퓨터로 부터 NC 프로그램의 수신
	중앙 컴퓨터로 현황 정보의 전송

제 3 절 중앙 컴퓨터의 모듈 개발

중앙 컴퓨터의 모듈들은 IPIS 시스템내에서 DNC의 기능들을 수행하기 위한 모듈들로 구성되어져, 위성 컴퓨터로 부터 전송되어오는 현황정보를 입력받아 공정의 진행 및 공작기계의 상태를 감시(Monitoring)하고 텍스트 또는 그래픽으로 출력해주며 생산 공정에 필요한 NC 프로그램의 관리 및 위성 컴퓨터로의 분배 기능을 수행하는 모듈들로 구성된다. NC 프로그램 관리 및 분배 모듈을 통해 구현된 중앙 컴퓨터의 기능들은 실시간 제어(Realtime Control)를 수행하기 위해 NC 프로그램 관리 및 분배 모듈들을 2개의 분리된 작업(Task)인 DNC-Master Task 와 PDA(Production Data Acqusition) Task로 나누어

DNC-Master Task에서는

- DNC Configuration
- NC 프로그램 관리
- NC 프로그램 분배
- 진도 관리

PDA Task에서는

- 현황 데이터 수집

의 모듈들로 구성하여 각 작업들간의 정보 전달은 메일박스(Mailbox)를 이용함으로써 비동기적으로 발생하는 사건들을 처리하였다.

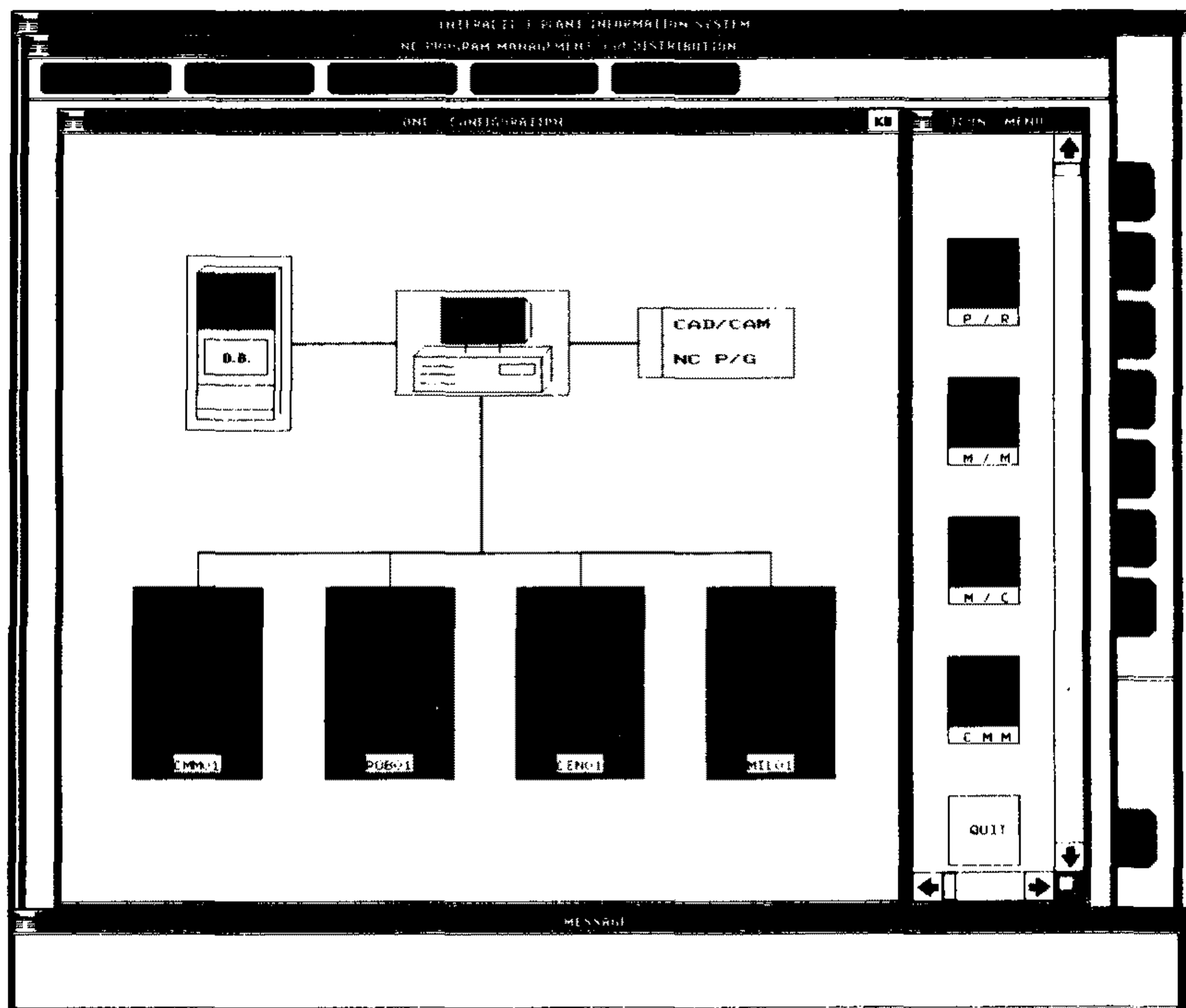
1. DNC Configuration 모듈

이 모듈에서는 현장에 설치되어 있는 각 기기들의 Layout를 심볼(Symbol)화된 아이콘(Icon)의 선택으로 사용자가 대화식으로 현장의 Layout를 설정하는 기능을 수행한다. 현장마다 각 공작기계들의 종류나 수가 다르고 동일 현장내에서도 작업 내용에 따라 공작기계의 변경이 생길 수 있으므로 이때마다 쉽게 갱신(Update)하는 작업을 지원함으로써 시스템의 변경에 유연하게 대처할 수 있다. 먼저 사용자가 키보드(Keyboard) 입력을 통해 현장에 설치되어져있는 네트워크 형태와 공작기계의 수를 입력하면 화면상에 시스템과 공작기계들이 위치할 빈 박스들이 입력되어진 수 만큼 네트워크 형태에 의해 연결되어진 화면으로 나타나고, 마우스(Mouse)를 이용하여 아이콘 메뉴에서 공작기계들을 선택하여 공작기계가 배치되어질 빈 박스를 선택하면 작업이 완료된다. 이 모듈을 통해 설정되어진 화면 구성은 현황관리 모듈에서 이용되어진다. <그림 5-5> 는 이 모듈의 화면 구성 예이다.

2. NC 프로그램 관리 모듈

NC 프로그램 관리 모듈에서는 생산 공정에서 필요한 NC 프로그램들을 관리하고 각 공작기계에 해당 NC 프로그램을 적절한 시기에 전송하기 위하여 필요한 데이터들을 입력받아 이들로 이루어진 NC 프로그램 File(이하, NC File이라 약함)을 구성한다 이 NC File에 대한 데이터의 변경, 데이터 레코드의 삭제, NC 프로그램의 정지(Blocking)등을 수행하기 위한 NC File 관리 모듈과

NC File의 데이터들을 해당 Key(ex, 오더 번호등)에 의해 NC 프로그램과 관련정보인, 오더, 공정, 공작기계등의 관계를 조회해 볼 수 있는 NC 프로그램 검색 모듈, 그리고 CAD/CAM 장비나 NC 프로그래밍 장치로 부터 생성된 NC 프로그램을 수령하는 모듈로 구성되어져 있다.



<그림 5-5> DNC Configuration 모듈의 화면구성

이 모듈을 본 년도에 개발된 Graphic Tool(ex, Multi-Window, List Box, Keyboard Input등)들을 이용하여 새로 재구성을 수행하였다.

가. NC File 관리

이 모듈에서는 NC 프로그램을 관리하기 위한 데이터들의 입력, 수정, 삭제, 정지를 수행하기 위해 4개의 서브 모듈로 구성된다.

(1) NC File 입력

CAD/CAM 장비등을 통해 생성되어진 NC 프로그램이 중앙 컴퓨터에 등록(Save) 되어진후 이를 관리하기 위한 관련 데이터를 입력하고자 할때에 이 모듈로 들어가게 된다. 이 모듈에서는 중앙 컴퓨터인 VAX-Station의 모니터에 그래픽 화면 구성을 통해 사용자가 직접 관련 데이터를 입력하도록 하였다. 이 모듈을 선택하면 NC File의 각 어트리뷰트(Attribute)들이 출력(Display)되고 각각의 위치에서 해당 데이터를 입력한후 레코드(Record) 형태로 저장한다. 화면에 출력된 어트리뷰트들 중에 실제 전송 날짜와 시간은 이때에 행해지지 않고 NC 프로그램 전송 모듈을 통해 실제 전송이 수행됨에 따라 갱신(Update) 된다.

(2) NC File 수정

작업 계획의 변경등에 따라 등록되어진 NC 프로그램의 관련 데이터를 변경하고자 할때는 이 모듈로 들어간다. 이 모듈에서는 Key Word로 NC 프로그램 번호를 입력하게 되고, 이를 통해 데이터 베이스에 해당 레코드가 있는지를 검사한다. 이를 만족하면

그에 해당하는 레코드의 어트리뷰트 값들이 화면상에 출력되고 각 어트리뷰트 값들을 순차적으로 변경한다. 새로이 개발된 Keyboard 입력 루틴을 적용함으로써 사용자의 편의를 도모하고자 했다.

The screenshot displays the 'INTERACTIVE PART INFORMATION SYSTEM' interface for 'NC PROGRAM MANAGEMENT & DISTRIBUTION'. It shows a list of attributes for an NC program, with some values redacted by black boxes. A 'MESSAGE' dialog box is open at the bottom, displaying the text 'Please Check your input data.' and a date stamp '1991 SEP 5 THU'.

Attribute	Value	Attribute	Value
NC PROGRAM NUMBER	P-CMM-NO11	NC PROGRAM CREATE DATE	91703719
NC PROGRAM SIZE	10300	PLANNED TRANSFER DATE	91704701
WORKSTATION NUMBER	CMM01	PLANNED TRANSFER TIME	16:30:00
PART NUMBER	P-2024-521	REAL TRANSFER DATE	
OPERATION NUMBER	P-CMM-0011	REAL TRANSFER TIME	
BLOCKING/RELEASE[Y/N]	N		
NC PROGRAM DESCRIPTION	COORD. MEASURING.		

MESSAGE
Please Check your input data.
.....

1991
SEP
5
THU

<그림 5-6> NC File 입력 모듈의 입력 화면

(3) NC File 삭제

공정의 완료나 삭제등에 의해 등록되어진 NC 프로그램의 관련 데이터를 데이터 베이스로부터 삭제하고자 할때 사용되어진다. Key Word로 NC 프로그램 번호를 입력하고 이를 통해 해당 레코드가 있는지를 검색한다. 이를 만족하면 해당 레코드의 어트리뷰트

값들을 화면상에 출력하고 사용자의 확인후에 작업을 완료한다.

(4) NC File 정지

등록되어진 NC 프로그램에 대한 접근(Access)을 제한하고자 할때는 NC File내의 정지(Blocking) 필드의 어트리뷰트 값을 "Y"로 주게 된다. 이 필드의 어트리뷰트 값만을 변경하고자 할때는 이 모듈을 선택하게 된다. 변경 모듈에서도 이 필드의 어트리뷰트 값들을 변경할수는 있지만 NC 프로그램에 대한 접근의 정지와 해제를 행한다는 기능의 특성상 따로 분리하였다.

나. NC 프로그램 조회

앞의 NC File 관리 모듈을 통해 구성되어진 NC File를 통해 사용자가 검색하고자하는 Key Word를 입력함으로써 NC 프로그램과 관련정보들을 조회하는 기능들로 구성되어져 있으며, 다음의 6개 서브 모듈들로 기능을 세분화 하였다. 이 모듈에서는 Multi-Window 와 List Box를 통한 Scroll 및 Panning 기능을 주어 사용자에게 더 많은 정보를 한 화면을 통해 제공하도록 구성하였다.

(1) 전체 NC 프로그램 조회

입력 모듈을 통해 등록되어진 전체 NC 프로그램 관련정보들을 NC 프로그램의 생성 날짜순으로 출력하는 기능을 제공한다. 출력

되는 정보로는 NC 프로그램 번호와 이에 관련된 오더, 공정, 공작기계 및 전송예정 날짜등을 테이블 형태로 출력한다.

(2) 스케줄(Schedule)된 NC 프로그램 조회

해당 작업 날짜에 현장의 공작기계로 전송되도록 스케줄된 NC 프로그램 관련 정보를 출력해보고자 할때는 이 모듈로 들어간다. NC 프로그램을 관리하기 위하여 구성된 NC File에 대하여 Key Word로 해당 작업 날짜를 입력값으로 하여 전 레코드를 검색한 후 전송 예정 날짜 및 시간 순으로 NC 프로그램 번호 및 관련 정보인 오더, 공정 및 공작기계등을 함께 출력한다.

출력되는 이러한 데이터들은 주어진 날짜에 수행되어질 공정과 공정에 할당된 공작기계, NC 프로그램 번호등을 함께 화면상에 출력함으로써 해당 날짜의 공정계획을 살펴볼 수 있고 NC 프로그램 전송 모듈에서 바로 이 모듈을 이용하여 전송 예정 시간 순으로 해당 공작기계에 NC 프로그램을 전송할수있다.

(3) 오더 번호별 NC 프로그램 조회

주어진 오더에 따라 전체 레코드를 검색해 오더에 관련된 공정의 공정 순서(Operation Sequence)로 각 공정에 할당된 NC 프로그램, 공작기계등을 출력해 보고자 할때 사용되어진다.

이 모듈에서는 주어진 Key Word로 오더 번호를 입력하여, 주어진 오더에 관련된 공정, NC 프로그램, 공작기계등을 출력함으로써 한눈에 하나의 오더가 수행되기 위해 거쳐야할 공정의 수와

순서, 각 공정에 할당되어진 공작기계, NC 프로그램등의 관계를 출력해볼수 있다.

INTERACTIZE PART INFORMATION SYSTEM
NC PROGRAM MANAGEMENT AND DISCUSSION

NC PROGRAM INDEX REFERENCE TABLE

NC Program Display

No	Operation	NC Program	M/C	PTD	B	NC Program
1	B-CEN-0081	B-CEN-NO81	CEN02	91/07/10	N	MACHINING CENTER
2	B-ROB-0082	B-ROB-NO82	ROB01	91/07/29	N	POLISHING ROBOT
3	B-CMM-0083	B-CMM-NO83	CMM02	91/07/30	Y	COORD. MEASURING

NC PROGRAM INDEX REFERENCE TABLE

NC Program Display

No	Operation	NC Program	M/C	PTD	B	NC Program
1	C-CEN-0031	C-CEN-NO31	CEN01	91/02/10	N	MACHINING CENTER
2	C-ROB-0032	C-ROB-NO32	ROB02	91/03/24	N	POLISHING ROBOT
3	C-CMM-0033	C-CMM-NO33	CMM01	91/06/28	N	COORD. MEASURING

1991
SEP
5
THU

<그림 5-7> 오더(Order) 번호에 따른 조회 화면

(4) 공정 완료 날짜별 NC 프로그램 조회

공정이 완료된 날짜를 Key Word로 전체 레코드를 검색하여 작업이 완료된 공정, 해당 공작기계, NC 프로그램등을 출력하는 기능을 제공한다. 공정 완료 날짜는 현장의 위성 컴퓨터를 통해 실시간으로 공정이 완료됨에 따라 On-Line으로 전송되어져 갱신되어 기록되고 완료된 공정에 대한 NC 프로그램에 대해서는 중앙 컴퓨터의 효율적인 메모리 사용을 위해 삭제할 수 있는 근거를

제시해 준다.

(5) 생성 날짜별 NC 프로그램 조회

NC 프로그램이 생성된 날짜를 Key Word로 하여 NC File내의 전 레코드를 검색해 NC 프로그램 번호와 관련정보를 출력하고자 할때는 이 모듈로 들어간다. 이 모듈을 통해 일정기간 동안 생성된 NC 프로그램을 조회해 볼수있다.

(6) 정지된 NC 프로그램 조회

NC File내의 전체 레코드중 Blocking 필드에 대해 Blocking된 NC 프로그램 번호만을 출력함으로써 접근이 제한된 NC 프로그램을 출력해 보고자할때는 이 모듈을 선택한다. 정지된 NC 프로그램의 허가(Release)시에 효율적인 검색기능을 제공한다.

다. NC 프로그램 등록

이 모듈에서는 CAD/CAM 장비나 NC 프로그래밍 장치를 통해 생성된 NC 프로그램을 전송매체(LAN)을 통해 중앙 컴퓨터의 메모리에 저장시키는 역할을 수행한다. CAD/CAM 장비와 중앙 컴퓨터와의 표준 Interface를 통해 NC 프로그램 및 NC 프로그램 관련 정보를 File 형태로 받아 NC 프로그램 관리 모듈의 입력 모듈을 통해 등록되어지도록 할 계획이다.

3. NC 프로그램 분배 모듈

중앙 컴퓨터(VAX-Station)에 등록되어진 NC 프로그램에 대하여 작업 스케줄에 따라 해당 위성 컴퓨터(IBM-PC)로 NC 프로그램을 전송하거나 각 위성 컴퓨터에서 작성되거나, 저장되어있는 NC 프로그램을 수신하는 기능을 수행한다. 2차년도에 수행한 NC 프로그램 분배 모듈에서는 상용 에뮬레이터인 Kermit를 이용하여 위성 컴퓨터를 중앙 컴퓨터의 터미널 모드 상태에서 위성 컴퓨터로부터 NC 프로그램을 전송 또는 수신하는 형태였으나 3차년도에서는 중앙 컴퓨터와 위성 컴퓨터에서 각기 통신 프로그램을 작성하여 중앙 컴퓨터가 Server의 역할을 함으로써 NC 프로그램을 On-Line으로 위성 컴퓨터로 전송/수신 하도록 하여 중앙 컴퓨터의 관리 및 제어 능력을 향상시켰다. 중앙 컴퓨터에서 작성된 통신 프로그램은 VAX-C 와 VMS System Service Routine들을 이용하여 위성 컴퓨터에서 작성된 통신 프로그램과 접속될 수 있도록 구성하였다.

가. NC 프로그램 전송

이 모듈은 중앙 컴퓨터에 등록되어 있는 NC 프로그램을 해당 위성 컴퓨터로 전송하는 기능을 수행한다. 이 모듈을 선택하면 시스템으로 부터 얻어진 낱자를 Key Word로하여 데이터 베이스내에 등록되어진 전체 NC 프로그램들에 대한 검색이 이루어진 후,

NC 프로그램, 공작기계, 공정등을 전송 예정 순으로 출력한다. 화면 구성은 3차년도에 개발된 List Box 기능을 이용하여 출력되는 정보들을 Scroll 하거나 Panning함으로써 많은 정보들을 보다 쉽게 조작할 수 있도록 구성하였다. 이와 같이 출력된 화면에서 전송하고자 하는 NC 프로그램 번호를 마우스로 선택하면 본 년도에 개발한 통신 프로그램을 통해 해당 위성 컴퓨터로 전송 되게 된다.

NC Program	Part No	Operation	M/C	PTD	PIT	
C-POB NO32	C 1308-203	C-ROB 0032	ROB02	91/03/24	14:30:00	PO
P-CMM NO11	P 2024-521	P-CMM 0011	CMM01	91/04/01	16:30:00	CO
P-ROB NO71	R-2793-267	P-ROB 0071	ROB01	91/04/10	11:00:00	PO
V-CMM NO82	V 4734-498	V-CMM 0082	CMM02	91/04/24	13:30:00	CO
C-CMM NO33	C 1308-203	C-CMM 0033	CMM01	91/06/28	15:30:00	CO
B-CEN NO81	B 1718-138	B-CEN 0081	CEN02	91/07/10	11:00:00	PO
B-ROB NO82	B 1718-138	B-ROB 0082	ROB01	91/07/20	10:30:00	PO
B-CMM NO83	B 1718-138	B-CMM 0083	CMM02	91/07/30	10:30:00	CO

<그림 5-8> NC 프로그램 전송 모듈의 화면 구성

나. NC 프로그램 수신

NC 프로그램 수신 모듈은 위성 컴퓨터에 있는 NC 프로그램을 중앙 컴퓨터에서 수신하는 기능을 수행하며 중앙 컴퓨터에서 수신하고자하는 위성 컴퓨터의 번호, NC 프로그램 번호, 중앙 컴퓨터에 등록하고자하는 NC 프로그램 번호를 입력하면 이러한 정보가 통신 프로그램쪽으로 넘어가 작업이 완료된다.

본 모듈에서 사용하고있는 중앙 컴퓨터인 VAX-Station은 2개의 터미날 포트만을 갖고있는 물리적인 제약때문에 2개의 위성 컴퓨터만을 연결하고있다. 다수의 위성 컴퓨터를 중앙 컴퓨터에 접속하기 위해 LAN(Ethernet)를 설치하여 이러한 Muti-Point Communication의 문제점들을 해결해 나가려하고 있다.

4. 현황관리 모듈

중앙 컴퓨터(VAX-Station)를 통한 현황정보의 수집과 수집된 현황정보에 의한 공정 및 공작기계의 상태(Status)를 관리하는 기능들로, 1차년도에 수행한 생산정보 수집 및 진도관리 모듈을 Update하여 구성한 모듈이다. 1차년도와 비교하여 본 연도에서 수행한 현황관리 모듈과의 차이점은 1차년도의 생산정보 수집 및 진도관리 모듈이 작업장의 현장 책임자에 의해 현황정보를 직접 Interactive하게 입력한 반면, 본 연도에서는 중앙 컴퓨터로부터 원거리에 위치한 현장의 위성 컴퓨터로부터 현황정보를 입력 받아 On-Line으로 중앙 컴퓨터에 전송하여 IPIS Database를 Update하게함으로써 실시간으로 현장의 상태를 중앙 관리실서 관

리(Monitoring)할 수 있도록 하였다. 또한 DNC Configuration 모듈을 통해 현장의 Layout에 따라 화면구성을 재구성하도록 하는 유연성을(Flexibility) 시스템에 제공하도록 구성하였다. 현황 관리 모듈을 선택하면 기본적으로 현장의 작업 상태를 그래픽으로 출력해주며 현장의 상태를 항상 실시간으로 Update하여 출력한다. 특정 공작기계나 현장 전체의 텍스트 출력은 진도관리 모듈을 통해 수행된다.

가. 진도 관리

이 모듈에서는 단위 공작기계별 또는 전체 현장내에서 수행해야 할 오더와 오더와 관련된 공정의 상태(Status)를 텍스트 윈도우(Window)로 출력하는 기능을 수행한다.

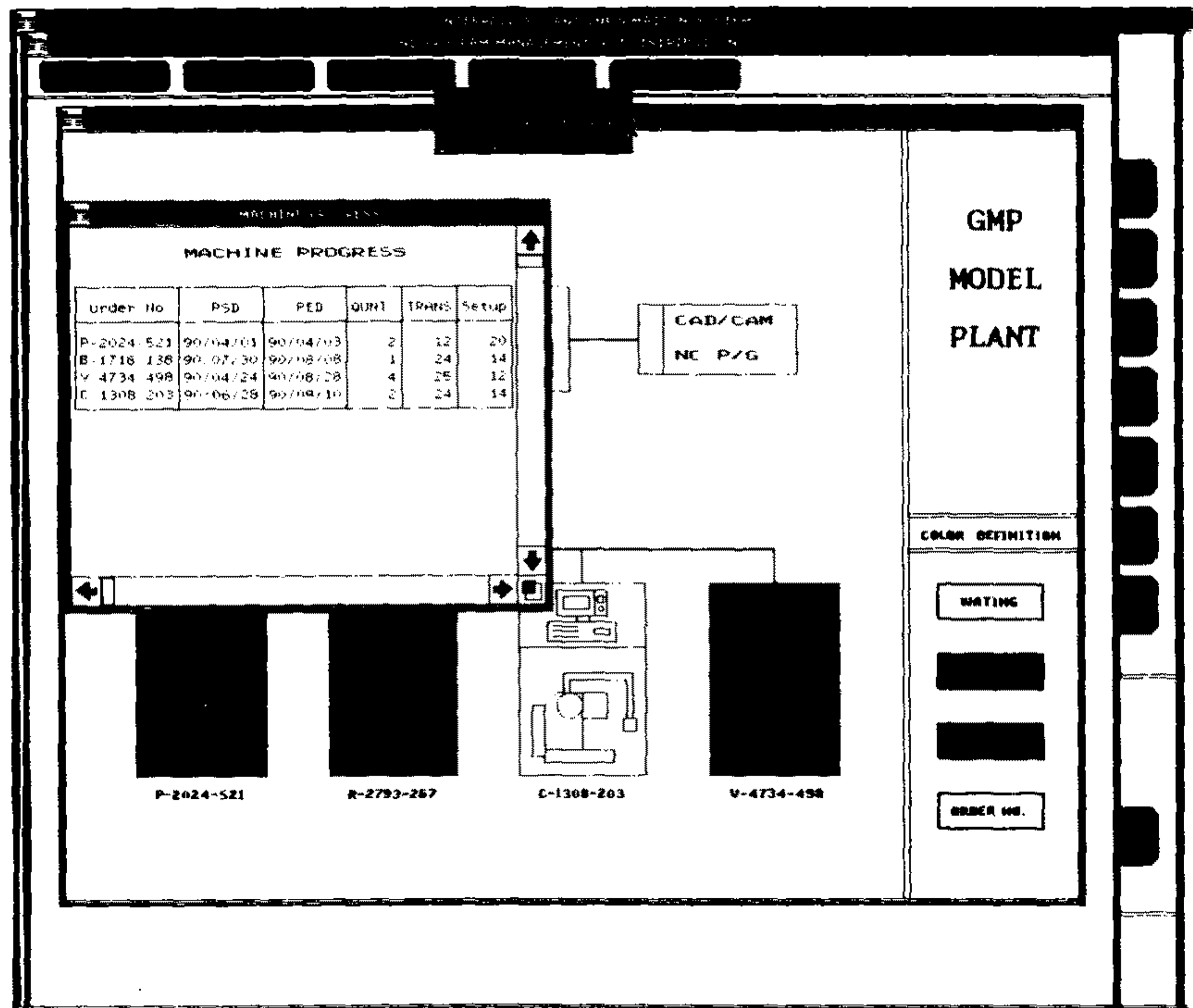
(1) 오더(Order) 조회 모듈

이 모듈은 현장내에 할당되어진 오더와 각 오더와 관련되어져 현재 또는 앞으로 수행해야 할 공정의 상태를 출력한다.

이 모듈을 선택하면, 우선 현장에서 수행해야 할 전체 오더와 오더관련 정보가 출력되고 이중 해당 오더를 선택하면 오더와 관련된 공정과 공정의 상태가 나타난다. 출력정보로는 오더 및 공정 번호와 작업 시작 예정 날짜 및 완료 날짜, 오더 및 공정의 상태를 출력한다. 사용자는 이 모듈을 통해 현장내에서 수행해야 할 각 오더와 공정의 진도관리를 수행할 수 있다.

(2) 공작 기계 부하 관리

이 모듈은 현장에 설치되어져있는 각 공작기계에서 수행해야 할 공정과 공정의 스케줄(Schedule)를 출력한다. 이 모듈을 선택한후 이미 화면상에 심볼화되어져 나타나 있는 공작기계들 중 조회해 보고자하는 공작기계를 선택하면 해당 공작기계에서 수행할 공정 번호와 계획된 작업 시작 및 완료 날짜가 출력된다.



<그림 5-9> 공작기계 부하 관리 모듈의 화면구성

나. 현황 정보 수집

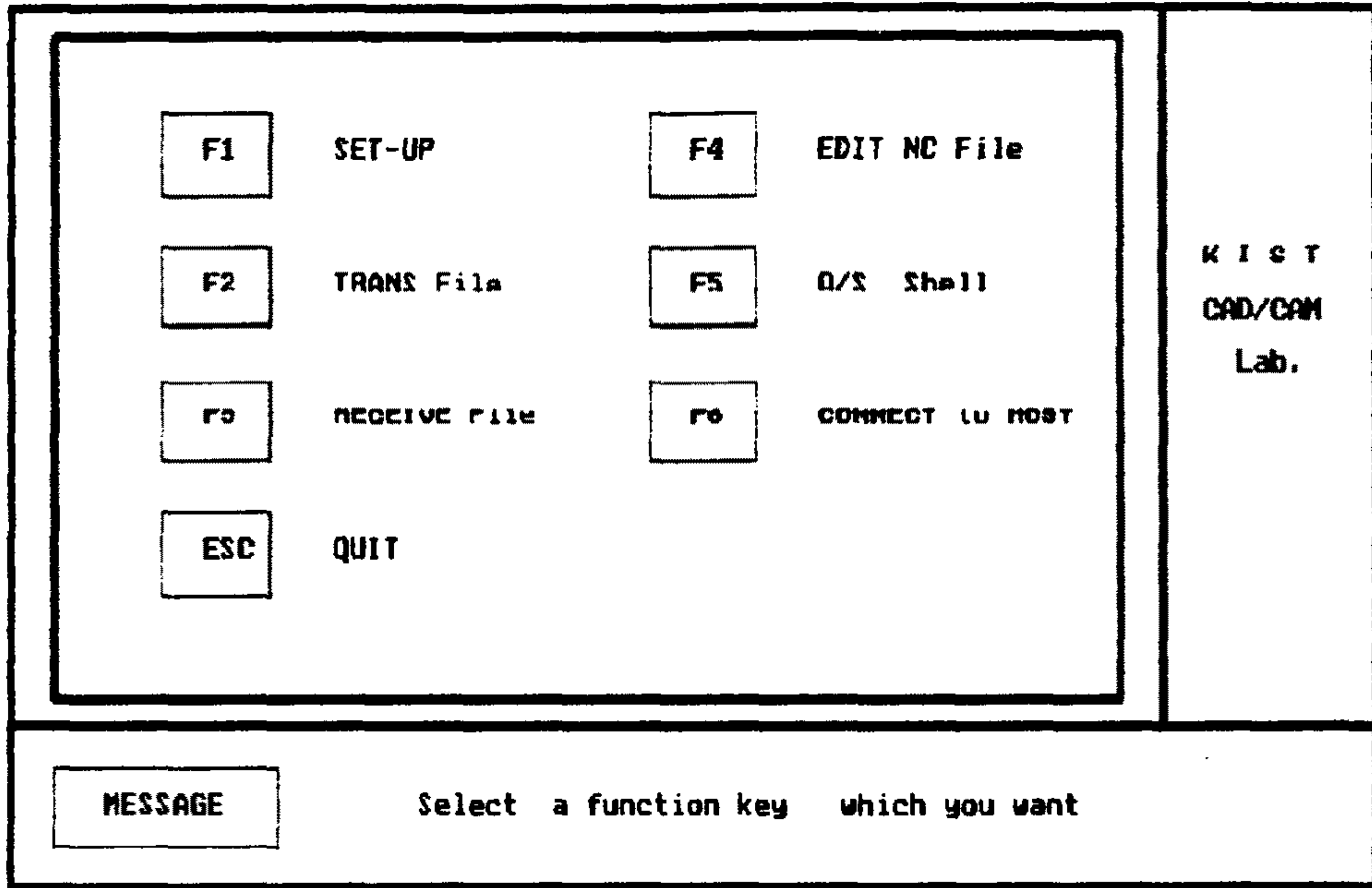
이 모듈에서는 위성 컴퓨터로부터 실시간으로 전송되어오는

현황정보를 입력 받아 IPIS 데이터 베이스를 갱신(Update)하는 기능을 수행한다. 이 모듈은 하나의 독립된 프로세스(PDA Task)로 비주기적으로 전송되어오는 현황정보를 입력받아 데이터 베이스에 저장하며, 메일박스를 통해 현황 정보를 DNC-Master Task로 보내줌으로써 그래픽 또는 텍스트로 현장을 모니터링(Monitoring)하도록 한다.

제 4 절 위성 컴퓨터의 모듈 개발

위성 컴퓨터의 소프트웨어는 현장에 있는 작업자가 위성 컴퓨터(IBM-PC)상에서 운영되는 메뉴(Menu)방식으로 구성된 화면을 통해 명령을 선택함으로써, 위성 컴퓨터로 부터 NC 프로그램의 CNC, ROBOT으로의 전송/수신, NC 프로그램의 수정, 현황정보의 입력 및 네트워크 공작기계 사이의 Interface를 수행하는 모듈들로 구성되어있다. 각 모듈의 수행은 PC상의 Function Key들을 사용하도록 하였으며, 각 모듈에서의 입력은 화면상에 출력되는 메뉴들을 선택하도록 하여 작업자의 Keyboard 입력을 최소화 하고 입력에서 발생할 수 있는 문제점의 발생 소지를 줄이도록 구성하였다.

<그림 5-10>는 위성 컴퓨터의 주 메뉴(Main Menu) 구성을 나타내는 화면이다.

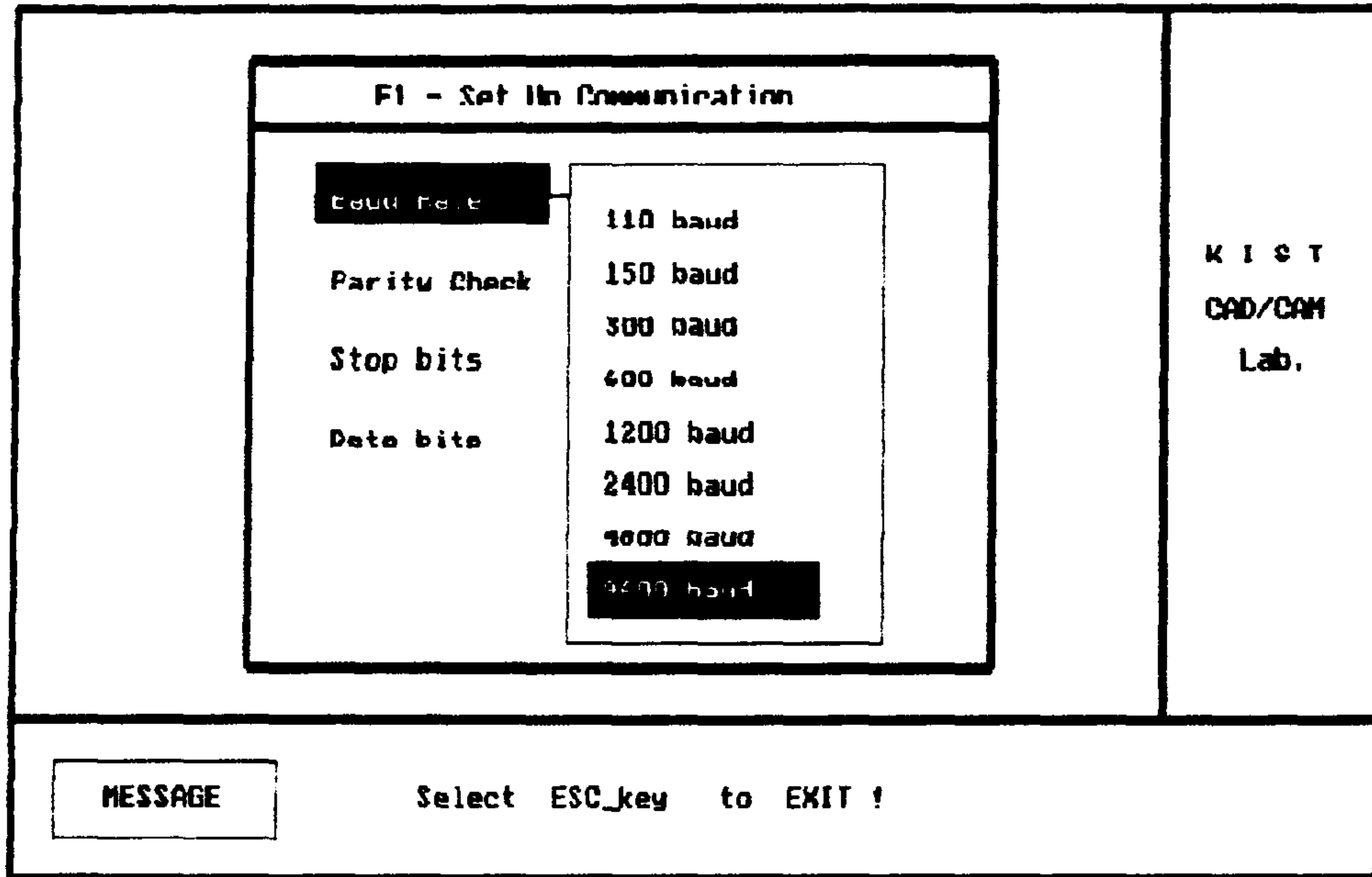


<그림 5-10> 위성 컴퓨터의 주 메뉴 화면구성

1. 통신 프로토콜 설정 모듈

이 모듈에서는 위성 컴퓨터에 접속된 CNC, ROBOT 등과 NC 프로그램을 전송 또는 수신하기 위하여 통신 프로토콜을 설정(Setup)하는 기능을 수행한다. 위성 컴퓨터에는 각기 다른 프로토콜을 가진 공작기계들이 접속될 수 있고 각 공작기계마다 통신 프로토콜을 재 설정하여야 하므로 이때마다 통신 프로그램을 수정하는 번거로움을 줄이고 작업자가 메뉴 방식으로 Interactive하게 통신 프로토콜을 설정하도록 하였다. <그림 5-11>은 통신 프로토

콜 설정 모드의 화면 구성 예이다.



<그림 5-11> 통신 프로토콜 설정 모듈의 화면구성

2. NC 프로그램 전송/수신 모듈

이 모듈에서는 중앙 컴퓨터로 부터 위성 컴퓨터로 전송 되어진 NC 프로그램을 위성 컴퓨터에 접속된 CNC, ROBOT 등의 Controller로 전송 또는 수신하는 기능을 수행한다. 위성 컴퓨터와 각 공작기계(CNC, ROBOT)와는 RS-232C로 연결하였으며 통신 프로토콜 설정 모드를 통해 공작기계와 접속되어진다. 이 모듈에서 수행되는 통신 프로그램은 Turbo-C의 그래픽 루틴들과 I/O 루틴들을 이용하여 작성하였으며, 메뉴를 선택한 후 전송 하고자

하는 NC 프로그램 번호를 Keyboard를 통해 입력함으로써 해당 동작기계로 전송하도록 구성하였다.

3. NC 프로그램 편집 모듈

이 모듈에서는 중앙 컴퓨터로 부터 전송되어온 NC 프로그램을 작업자가 일부 수정하거나 새로운 NC 프로그램을 작성하고자 할 때 작업자에게 NC 프로그램을 작성/편집할 수 있는 기능을 제공한다. 위성 컴퓨터의 전체 화면 구성은 그래픽으로 구성되어져 있어 이 모듈에서는 이미 상용화되어 있는 Utility를 이용하여 NC 프로그램을 작성 또는 편집하도록 하였으며 그래픽 메뉴상태에서 작업자가 수행하고자 하는 NC 프로그램 번호를 입력함으로써 그래픽 모드로 부터 빠져나와 텍스트 모드상태에서 작업을 수행하도록 구성하였다. 이 모듈에서 사용한 Utility는 Personal Editor를 사용하였으며 작업이 완료되면 다시 그래픽 상태의 메뉴 구성으로 전환되도록 하였다.

4. 중앙 컴퓨터와의 접속 모듈

이 모듈은 중앙 컴퓨터로 부터 NC 프로그램을 수신하기 위한 수신 프로그램, 현황 정보를 전송하기 위한 전송 프로그램, 그리고 메뉴 방식에의해 위성 컴퓨터로 부터 입력된 현황 정보를 중앙 컴퓨터로 전송하기 위한 데이터 포맷으로 바꾸는 세가지 기능

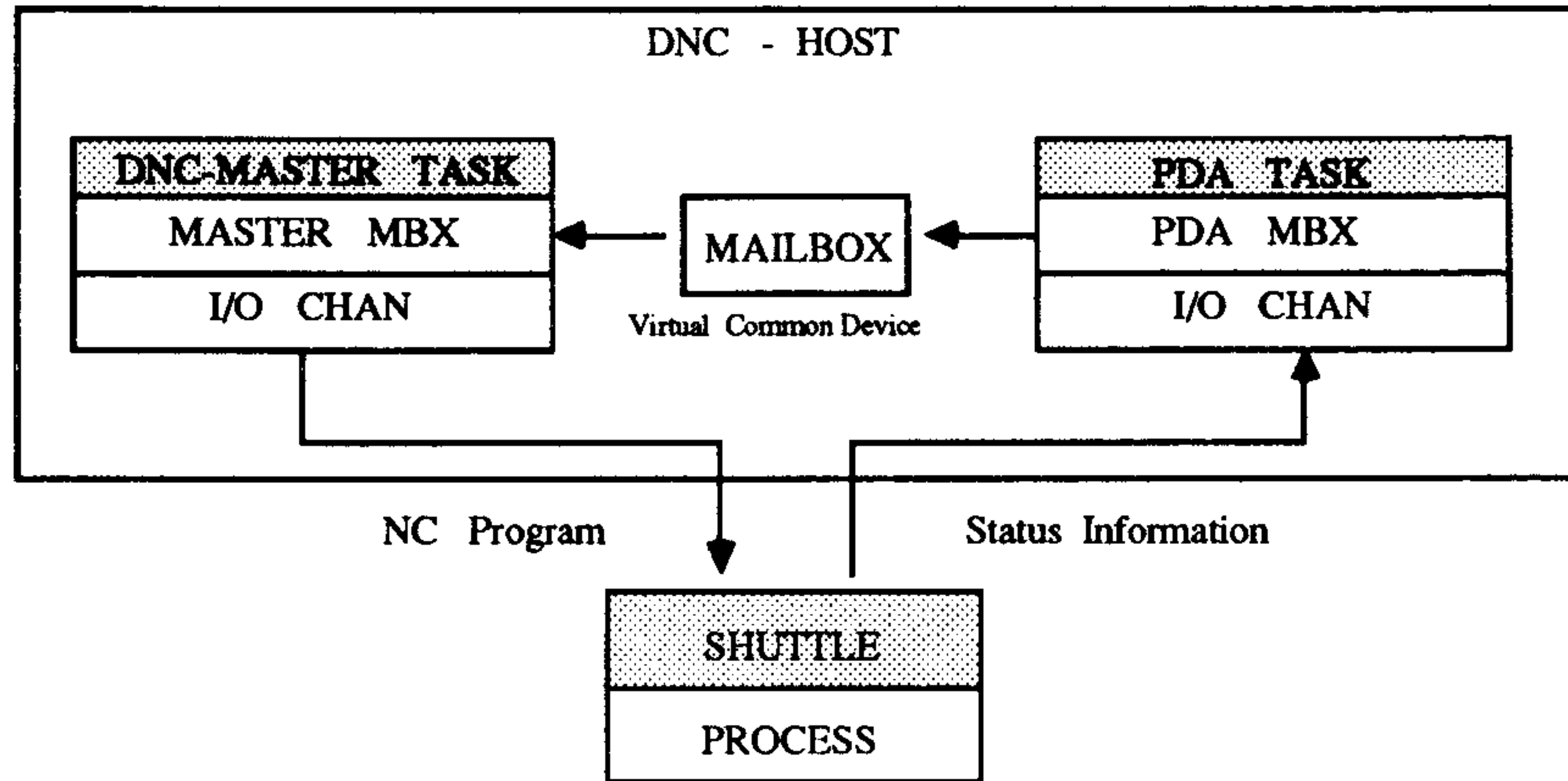
들로 구성된다. 중앙 컴퓨터와 NC 프로그램을 수신하거나 현황 정보를 전송하기 위한 통신 프로그램은 Turbo-C의 I/O Interface 루틴들을 이용하여 작성하였으며 공정의 상태를 공정의 시작 (Begin), 완료(End), 중단(Breakdown), 중단 완료(Breakdown End)의 네가지로 구분하여 공정의 시작 과 완료, 중단 발생 및 원인 등의 현황정보를 실 시간으로 중앙 컴퓨터로 전송하도록 구성하였다. <그림 5-12>는 현황 정보를 입력하는 화면 구성의 예이다.

FK - Production Data Acquisition	
MACHINE ID : CEND1	
JOB TYPE : BEGIN END <input checked="" type="checkbox"/> D/T <input type="checkbox"/> D/N	
Trouble No. : 3	
<ul style="list-style-type: none"> 1. Tool problem 2. Fixture problem 3. Machine problem 4. Material lack 5. NC program error 6. Operation error 	
Actual Quantity : 01	
Expected Trouble End Date : 91/04/09	
K I S T CAD/CAM Lab.	
MESSAGE	Would you send data to DNC HOST : [Y/N] _

<그림 5-12> 현황정보 전송 화면구성

제 5 절 중앙 컴퓨터와 위성 컴퓨터간의 정보연결

NC 프로그램 관리 및 분배 모듈에서는 IPIS 시스템에 연결된 다수의 공작기계들을 실시간 제어 관리하기 위하여 멀티 태스킹이 충족되어야 하고, 독립적인 작업들이 수행되면서도 각 작업들간에 효율적인 정보전달이 이루어져야 한다. 다작업 실시간 제어 시스템의 중요 요소는 전언(Message)를 메일박스(Mailbox)를 통하여 전달받을 수 있는 분리된 작업들로 나눌수있는 점이다. 다작업 실시간 제어 시스템은 소프트웨어 기능을 대화 가능한 분리된 작업(프로세스 또는 프로그램)으로 나눔으로 더욱 확장될 수 있고 이렇게 함으로써 비동기(Asynchronous)적으로 일어나는 사건들을 다룰 수 있게 된다. NC 프로그램 관리 및 분배 모듈 내에서의 정보의 흐름은 크게 2가지로 대별되는데 중앙 컴퓨터내에서의 작업들간의 통신은 Virtual Common Device인 메일박스를 이용하고, 중앙 컴퓨터와 현장의 위성 컴퓨터간의 통신은 Physical Link를 통한 I/O Port를 이용하여 NC 프로그램 및 현황 정보를 송수신 할수있게 구성하였다<그림 5-13>. 중앙 컴퓨터에서 다작업 실시간 제어를 위해 개발된 소프트웨어는 VAX-C 언어로 쓰여졌으며, VMS 운용 시스템과 시스템 서비스 루틴들이 사용되었다.



<그림 5-13> Mailbox 와 I/O Port를 이용한 정보연결

1. 비동기 시스템 루틴

VMS 운용 시스템에서는 어떤 특별한 사건이 발생하는 경우 프로그램의 수행이 중단되도록 요구될 때가 있다. 이러한 인터럽트 (Interrupt)는 프로그램이 수행될 때 특정 시간에 순차적으로 발생하는 것이 아니기 때문에 비동기 시스템 루틴(AST)이라고 부르며, 여기서 사건을 다룰 사용자가 지정한 부프로그램이 수행된다. AST를 설정하는 시스템 서비스는 어떠한 사건이 일어날때 수행될 부프로그램의 주소가 하나의 Argument로 되어 요청된다.

2. 동기 신호기

실시간 제어 상황에서는 어떠한 사건들의 동기화가 요구되게 된다. 동기 신호기는 시스템 소프트웨어에서 사건의 순서를 제어하게 되며, VMS 운용 시스템에서는 이벤트 플래그를 조작하기 위한 시스템 서비스 루틴들을 제공한다. 어떤 특정 이벤트 플래그를 설정(Set), 취소(Clear) 또는 읽어(Read)들임으로써 사건의 발생이나 완료, 프로세스간의 동기화를 기할 수 있다.

3. Input/Output 서비스

가. 프로세스(Process)와 디바이스(Device)간의 Channel 할당

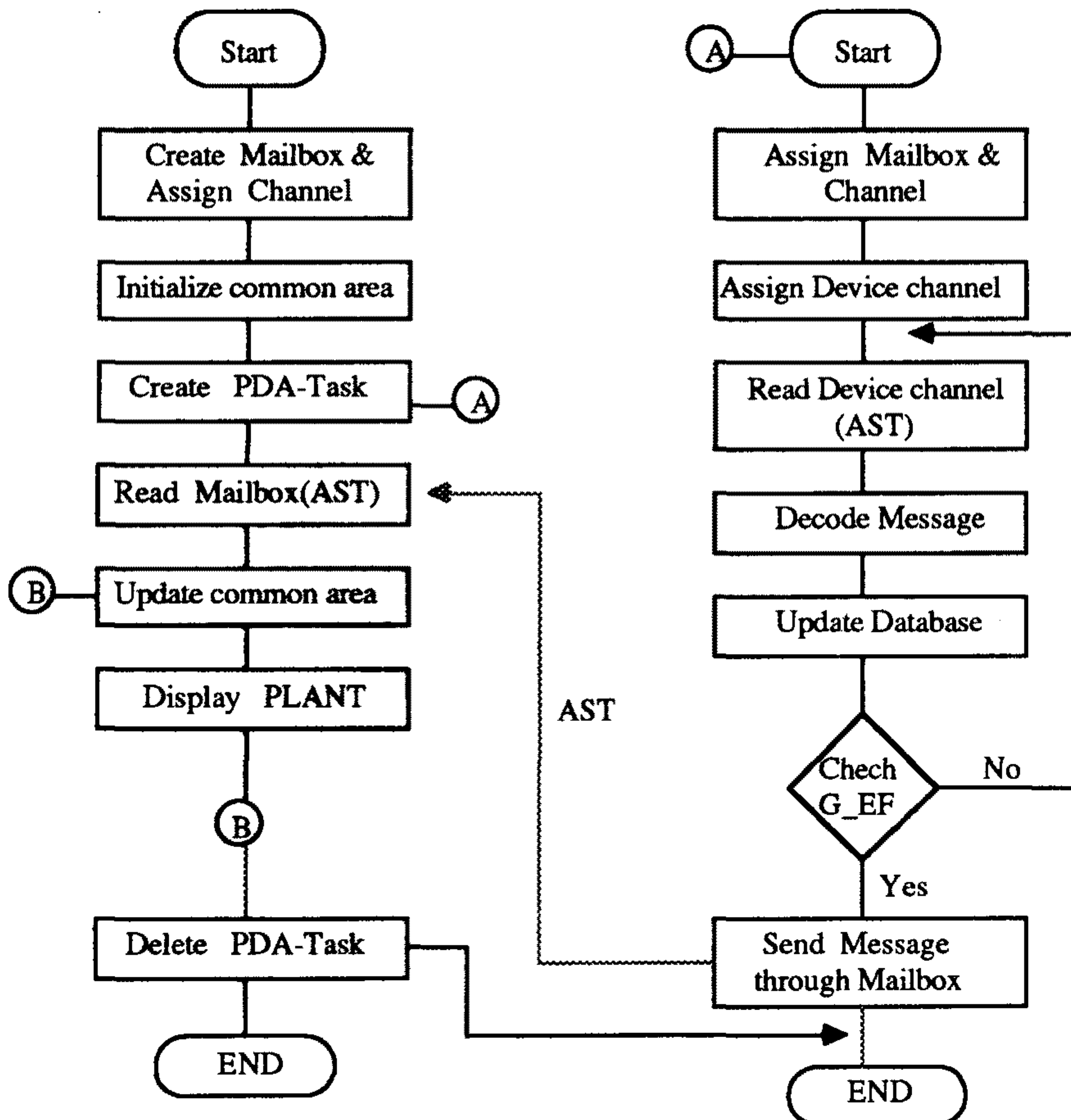
Physical Device상에서 Input 또는 Output이 수행되기전에 프로세스와 디바이스간에 Path를 제공하기 위하여 디바이스에 Channel를 할당해야만 한다. VMS에서는 Assign I/O Channel 시스템 서비스를 통해 프로세스와 디바이스간의 Path를 형성한다.

나. 프로세스(Process)간의 Channel 할당

메일박스는 프로세스간의 통신을 위하여 사용되어지는 가상의 디바이스이다. Create mailbox 와 Assign Channel 시스템 서비스를 통해 하나의 프로세스가 메일박스를 만들어 그곳에 Channel를 할당시키고, 또다른 프로세스들은 그 메일박스의 Channel에 할당할 수 있다.

다. I/O 의 수행

VMS에서 모든 Input과 Output은 Queue I/O Request 시스템 서비스를 통해 이루어진다. QIO 시스템 서비스에서 요구되는 Argument는 가, 나, 를 통해 얻은 Channel 번호와 메일박스에 전언을 쓰거나 읽을것을 지정하는 Function Code 및 AST 부프로그램이 들어가게 된다.



<그림 5-14> 중앙 컴퓨터에서 작업(Task)간의 Flow

제 6 장 Basic Database Management Routine 개발

제 1 절 BDMR의 역할

1. BDMR의 필요성

BDMR(Basic Database Management Routine)이란 플랜트 통제 시스템을 개발하면서 Program의 Integration을 높이기 위해 각 Programmer들의 요구에 의해서 만들게 된 Routine이다.

BDMR이란 Routine명이 의미하듯이 Database를 Handling하는데 기본적으로 필요한 입력(Input), 삭제>Delete), 수정(Change), 출력(Display)을 위한 Standard Function이다. 즉, Database를 Handling하는 Program을 작성하고자 하면 BDMR의 Function을 부르기만 하면 된다. 아울러 그 Function들이 수행되기 위해선 그 내용에 따라 몇개의 Argument를 필요로 하며 이때 Programmer는 필요로 하는 조건에 따라 해당 Argument에 적당한 값만 Assign하면 된다.

이렇게 함으로써 플랜트 통제 시스템에서 사용하는 Database인 Relational Database의 구축 및 유지가 용이해지고 Database 관리에 일관성을 추구할 수 있다. 또한, 유사한 Function들을 Standard화 함으로써 Program 개발시 발생할 수 있는 중복 투자를 배제할 수 있다. 그리고, Program Size를 대폭적으로 줄이고 Compile 시간 또한 감소시킬 수 있다. 뿐만 아니라 Error 검출

및 그 Code화도 용이하다는 장점이 있다.

2. IPIS와의 연계 방안

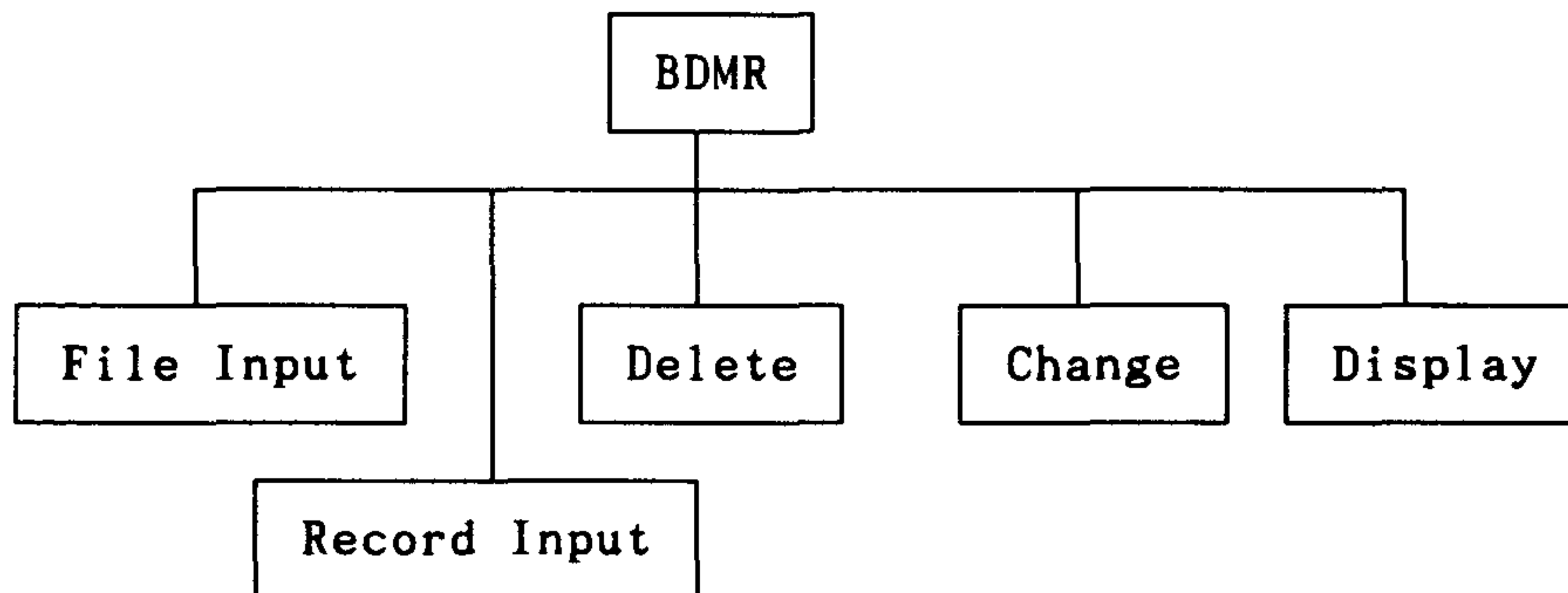
앞 절의 BDMR의 필요성에서 잠깐 언급했듯이 BDMR은 Standard Function들의 집합이기 때문에 각 Programmer들이 Program을 개발하는데 있어서 Database Handling이 필요할 때마다 필요한 Function을 불러서 사용하기만 하면 된다. 다만, 이 Function이 필요로 하는 Argument들을 각 Programmer들에게 명시하여 그 형태에 맞춰서 사용할 수 있도록 정의를 명확히 해 주어야 한다. 이렇게 함으로써 각 Programmer들은 Database Handling 부분에 대해서는 신경을 쓰지 않아도 된다.

제 2 절 BDMR의 구성 및 기능

1. BDMR의 구성

가. BDMR의 VTOC(Visual Table Of Contents)

BDMR은 입력(화일 입력 · 레코드 입력), 삭제, 수정, 출력의 기능을 수행할 수 있도록 구성되어 있다. BDMR의 VTOC(Visual Table Of Contents)는 다음과 같다.



<그림 6-1> BDMR의 VTOC

나. BDMR의 사용 방법

BDMR의 형태는 다음과 같다.

```

rdb_func(code, rse, mod_arg, funct, functprm)

```

위 Function 에서 사용한 각 Argument들을 설명하면 다음과 같다.

(1) Code

Database를 Handling할 때, 수행하고자 하는 기능에 대한 Argument이다. 즉, 입력, 삭제, 수정, 출력을 의미하며 아래와 같이 Code화 시켰다.

```

#define STR_REC 0
#define DEL_REC 1

```

```
#define MOD_REC 2
#define GET_REC 3
```

(2) Rse

하고자 하는 기능을 저장하는 Struct Type의 Data로 Rdb의 Record Selection Expression(RSE)에 해당하는 정보로 구성된다.

```
struct
{
    int first;
        /* 조건을 만족하는 Stream중 처음 나오는
        Record들의 지정된 갯수 */
    int no_fld, field_key[FLD_NO];
        /* 기능을 수행하고자 할 때 조건이 되는
        Field의 갯수와 그 Field 명 */
    union def_type stt_range[FLD_NO],
        end_range[FLD_NO];
        /* 기능을 수행하고자 할 때 조건이 되는
        Field의 범위 */
} rse;
```

(3) Mod_arg

Record를 수정하고자 할때 필요한 정보이다. 각각을 설명하면 다음과 같다.

```

struct
{
    int    no_fld;
           /* 수정하고자 하는 Field의 갯수 */
    int    field_key[FLD_NO];
           /* 수정하고자 하는 Field 명 */
    union def_type value[FLD_NO];
           /* 수정하고자 하는 Field 값 */
} mod_arg;

```

(4) Funct

Programmer가 정의한 User Function을 의미한다. 이 부분에 대한 작성은 Function을 불러서 사용하는 Programmer들 각자가 해야 한다.

(5) Functprm

위의 User Function에서 필요로 하는 Argument를 의미한다. 이 부분에 대한 작성도 앞의 User Function과 마찬가지로 Programmer 각자가 해야 한다.

다음은 BDMR을 사용하는 방법에 대한 간단한 예제이다. Argument중에서 Code를 STR_REC, DEL_REC, MOD_REC, GET_REC중에서 어떤 것을 사용하느냐에 따라 입력, 삭제, 수정, 출력을 수행한다.

<표 6-1> BDMR의 사용 방법

	BDMR을 부르는 방법(예)	설 명
입 력	rdb_func(STR_REC, ODR, &odr, 0, 0)	Order Relation을 저장
삭 제	rdb_func(DEL_REC, &rse, 0, 0, 0)	해당 조건에 맞는 Record를 삭제
수 정	rdb_func(MOD_REC, &rse, &mod_arg, 0, 0)	해당 조건에 맞는 Record를 수정
출 력	rdb_func(GET_REC, &rse, 0, 0, 0)	해당 조건에 맞는 Record를 출력

2. BDMR의 기능 및 적용 예

BDMR의 기능을 열거 하기 전에, 이해를 돕기 위해서 BDMR을 이용한 Database Handling Program의 출력 결과를 가지고 설명하려 함을 미리 언급 한다. 6장에서 예제로 사용한 Relation은 Order Relation과 Operation Relation이다.

현재는, 대표적인 Relation인 Order Relation과 Operation Relation만 적용할 수 있는 상황이지만 앞으로는 플랜트 통제 시스템에서 Handling하는 Database를 구성하는 모든 Relation에 대해서 적용할 수 있는 BDMR을 개발할 계획이다. 또한, BDMR을 이용한 독자적인 Database Management Module도 개발할 계획이다.

가. 저장

(1) Batch 방식의 저장

Database내의 Data를 Handling하기 위해선 Database내에 Data들이 존재해야 한다. 그러므로 Database를 처음 구축한 다음에 Data File로 부터 Data를 읽어 들어서 Database에 Loading 하는 기능이 필요하다. Batch방식의 저장 기능이 바로 이 기능이다. 다음은 앞으로 예제에서 사용할 Database에 Data를 저장하기 위한 Data File의 List이다.

```
O TEL041 Upper Clamping Plate
  CLM-01 91/06/12 91/06/25 1 4 1 C
    P 1 MIL01 91/06/12 91/06/12 91/06/12 91/06/13
      01:00 05:00 01:00 1 4 N N N 0 0 C
      .
      .
      .
O TEL042 Lower Clamping Plate
  CLM-03 91/06/19 91/07/03 1 4 1 C
      .
      .
      .
O TEL045 Ejector Remainer Plate
  EJR-12 91/06/28 91/07/05 1 4 1 Q
    P 1 MIL01 91/06/28 91/06/28 91/06/28 91/06/28
      01:00 03:00 01:00 1 4 N N N 0 0 C
      .
      .
```

위의 Data File 을 이용해서 Database에 Data를 저장했을때의 Order Relation과 Operation Relation의 내용들을 List해보면 다

음과 같다.

```
( Order Relation ----- )
Order = TEL041, Comp = CLM-01, PSD = 91/06/12, PLS = 1
Order = TEL042, Comp = CLM-03, PSD = 91/06/19, PLS = 1
Order = TEL043, Comp = CAV-06, PSD = 91/06/26, PLS = 1
Order = TEL044, Comp = CAV-04, PSD = 91/07/03, PLS = 1
Order = TEL045, Comp = EJL-12, PSD = 91/06/28, PLS = 1
```

```
( Operation Relation ----- )
Order = TEL041, Oper = 1, Mach = MIL01, PSD = 91/06/12
Order = TEL041, Oper = 2, Mach = DRL01, PSD = 91/06/14
Order = TEL041, Oper = 3, Mach = CEN01, PSD = 91/06/17
Order = TEL041, Oper = 4, Mach = CMM01, PSD = 91/06/20
Order = TEL041, Oper = 5, Mach = GRN01, PSD = 91/06/21
Order = TEL041, Oper = 6, Mach = ROB01, PSD = 91/06/22
.
.
.
Order = TEL045, Oper = 1, Mach = MIL01, PSD = 91/06/28
Order = TEL045, Oper = 2, Mach = DRL02, PSD = 91/06/29
Order = TEL045, Oper = 3, Mach = EDM02, PSD = 91/06/30
Order = TEL045, Oper = 4, Mach = ROB01, PSD = 91/07/02
Order = TEL045, Oper = 5, Mach = CMM03, PSD = 91/07/05
```

앞으로는 위의 Database 상황하에서 레코드 저장, 삭제, 수정, 출력의 기능들을 설명하겠다.

(2) 대화형 방식의 저장

Database내에 새로운 정보를 저장하는 이 기능은 각 Relation 별로 이루어질 수 있다. 다음 예제를 가지고 설명해 보면 현재 Database내에는 5개의 Order가 저장되어 있는데 새로운 Order를 입력 받는 경우에, 새로 입력 받은 Order명이 이미 존재하는

Order명과 일치하면 "Duplicates are not allowed."라는 Error Code를 반환해 주는 것과 더불어 다시 Order명을 입력 받게 한다. Order명이 중복되지 않게 하려면 Database를 정의하는 DDL(Data Definition Language)를 사용해서 정의를 하면 된다. 아래의 예제를 보면 Order TEL041 을 입력하면 Error Message가 출력된다. 왜냐 하면 그 Order명은 이미 존재하는 Order명이기 때문이다.

```
* Relation      : ODR
* Please input the data.
Order :   ODN      CPN      PSD      PED      PLS      CODE
        TEL041  UCP-01  91/06/13  91/06/26   1       Q
```

* Fail : Duplicates are not allowed.

```
Order :   ODN      CPN      PSD      PED      PLS      CODE
        TEL046  EJC-13  91/06/29  91/07/05   1       Q
Operation :  MCN      PSD      PED      SUT      PLQ      CODE
            MIL03  91/06/29  91/06/29  01:00     1       C
Operation :  MCN      PSD      PED      SUT      PLQ      CODE
            DRL03  91/06/30  91/06/30  02:00     1       Q
!! Success : Record input is complete.
```

위와 같이 입력한 후 Order Relation 의 내용과, Operation Relation에서 Order TEL046의 공정(Operation)들의 내용을 출력해 보면 다음과 같다.

```
( Order Relation ----- )
Order = TEL041, Comp = CLM-01, PSD = 91/06/12, PLS = 1
Order = TEL042, Comp = CLM-03, PSD = 91/06/19, PLS = 1
Order = TEL043, Comp = CAV-06, PSD = 91/06/26, PLS = 1
```

```
Order = TEL044, Comp = CAV-04, PSD = 91/07/03, PLS = 1
Order = TEL045, Comp = EJL-12, PSD = 91/06/28, PLS = 1
Order = TEL046, Comp = EJC-13, PSD = 91/06/29, PLS = 1
```

```
( Operation Relation ----- )
```

```
Order = TEL046, Oper = 1, Mach = MIL03, PSD = 91/06/29
Order = TEL046, Oper = 2, Mach = DRL03, PSD = 91/06/30
```

다음은 한 Order에 대해서 여러 가지 공정(Operation)이 있는 데 임의의 공정을 삽입하는 경우에 대한 예제이다.

```
* Relation      : OPR
* Please input the data.
Operation : ODN   MCN   PSD       PED       SUT   PLQ CODE
           TEL046 EDM01 91/07/01 91/07/02 01:00   1     0
!! Success : Record input is complete.
```

위와 같이 입력한 후 Operation Relation 에서 Order TEL046의 공정의 내용을 출력 해 보면 다음과 같다.

```
( Operation Relation ----- )
```

```
Order = TEL046, Oper = 1, Mach = MIL03, PSD = 91/06/29
Order = TEL046, Oper = 2, Mach = DRL03, PSD = 91/06/30
Order = TEL046, Oper = 3, Mach = EDM01, PSD = 91/07/01
```

나. 삭제

Database내의 Record들을 삭제하는 이 기능은 Relation별로 이루어 진다. 다음은 그에 대한 예이다. 우선 Order Relation에 대한 예이다.

* Relation : ODR
* Order number : TEL046

위와 같이 Order TEL046을 삭제시킨 다음 Order Relation의 내용을 출력시켜 보면 다음과 같다.

```
( Order Relation ----- )  
Order = TEL041, Comp = CLM-01, PSD = 91/06/12, PLS = 1  
Order = TEL042, Comp = CLM-03, PSD = 91/06/19, PLS = 1  
Order = TEL043, Comp = CAV-06, PSD = 91/06/26, PLS = 1  
Order = TEL044, Comp = CAV-04, PSD = 91/07/03, PLS = 1  
Order = TEL045, Comp = EJR-12, PSD = 91/06/28, PLS = 1
```

또한 Operation Relation에서 Order TEL046에 해당하는 공정들의 내용을 출력시켜 보고자 하면 다음과 같은 Error Code를 반환한다.

* Order number : TEL046

* Fail : There are no operations with that order number.

다음은 Operation Relation에 대한 예제이다. 임의의 공정을 삭제하기 전의 Order TEL045의 Operation Relation의 내용은 다음과 같다.

```
( Operation Relation ----- )  
Order = TEL045, Oper = 1, Mach = MIL01, PSD = 91/06/28  
Order = TEL045, Oper = 2, Mach = DRL02, PSD = 91/06/29  
Order = TEL045, Oper = 3, Mach = EDM02, PSD = 91/06/30
```



```
Order = TEL045, Oper = 4, Mach = ROB01, PSD = 91/07/02
Order = TEL045, Oper = 5, Mach = CMM03, PSD = 91/07/05
```

다음과 같이 Order TEL045 의 공정 중 첫번째 공정을 삭제하여 Operation Relation의 내용을 출력시켜 보면 아래와 같다.

```
* Relation          : OPR
* Order number      : TEL045
* Operation number  : 1
```

```
( Operation Relation ----- )
Order = TEL045, Oper = 1, Mach = DRL02, PSD = 91/06/29
Order = TEL045, Oper = 2, Mach = EDM02, PSD = 91/06/30
Order = TEL045, Oper = 3, Mach = ROB01, PSD = 91/07/02
Order = TEL045, Oper = 4, Mach = CMM03, PSD = 91/07/05
```

임의의 Record를 삭제하고자 할때 존재하지 않는 Record이면 BDMR을 부른 Function에 Error Code를 반환해 주어서 그에 따른 적절한 대응을 BDMR을 사용한 Programmer가 할 수 있도록 하였다.

다. 수정

Database내의 각 Field를 수정하는 이 기능은 Relation별로 이루어 질 수 있다. Record 전체를 수정하거나 아니면 Field 몇 개만 수정할 수 있다. 이때 각 Field들을 Code화하여 Standard Function으로 수행한다. 다음은 Order Relation에 대한 수정의 예이다.

```

* Relation      : ODR
* Order number  : TEL042
. The number of change field : 3
. Change field name           : CPN
. Change field value          : LCP-03
. Change field name           : PSD
. Change field value          : 91/06/20
. Change field name           : PLS
. Change field value          : 5
!! Success : Record change is complete.

```

```

( Before changing )
Order = TEL042, Comp = CLM-03, PSD = 91/06/19, PLS = 1

```

```

( After changing )
Order = TEL042, Comp = LCP-03, PSD = 91/06/20, PLS = 5

```

다음은 Operation Relation에 대한 예제이다.

```

* Relation      : OPR
* Order number  : TEL044
* Operation number : 2
. The number of change field : 3
. Change field name           : MCN
. Change field value          : CCC02
. Change field name           : PSD
. Change field value          : 91/07/07
. Change field name           : PLQ
. Change field value          : 3
!! Success : Record change is complete.

```

```

( Before changing )
Order = TEL044, Oper = 2,
Mach = DRL02, PSD = 91/07/05, PLQ = 1

```

```

( After changing )
Order = TEL044, Oper = 2,
Mach = CCC02, PSD = 91/07/07, PLQ = 3

```

다음과 같이 임의의 Record를 수정하고자 할때 존재하지 않는 Record이면 BDMR을 부른 Function에 Error Code를 반환해 주어서

그에 따른 적절한 대응을 BDMR을 사용한 Programmer가 할 수 있도록 하였다.

* Order number : TEL046

* Fail : There are no orders with that code.

라. 출력

Programmer가 정의한 조건에 맞는 Data들을 출력하는 이 기능은 한개 이상의 Sort Key를 부여해서 Sorting도 수행할 수 있다. 예를 들면, Planned Start Date순으로 Order들의 List를 본다든지 아니면 A라는 날짜부터 B라는 날짜까지의 Operation들의 List를 볼 수 있는 것 등을 들 수 있다. 다음은 Planned Start Date순으로 Order들을 List해 본 것이다.

```
( Order Relation ----- )
Order = TEL041, Comp = CLM-01, PSD = 91/06/12, PLS = 1
Order = TEL042, Comp = CLM-03, PSD = 91/06/19, PLS = 1
Order = TEL043, Comp = CAV-06, PSD = 91/06/26, PLS = 1
Order = TEL045, Comp = EJL-12, PSD = 91/06/28, PLS = 1
Order = TEL044, Comp = CAV-04, PSD = 91/07/03, PLS = 1
```

다음은 Planned Start Date가 A와 B사이에 있는 Order들의 List이다.

```
* Please input the range of data.
A : 91/06/01
B : 91/06/15
```

(Order Relation -----)
Order = TEL041, Comp = CLM-01, PSD = 91/06/12, PLS = 1

* Please input the range of data.

A : 91/06/16

B : 91/07/31

(Order Relation -----)
Order = TEL042, Comp = CLM-03, PSD = 91/06/19, PLS = 1
Order = TEL043, Comp = CAV-06, PSD = 91/06/26, PLS = 1
Order = TEL045, Comp = EJR-12, PSD = 91/06/28, PLS = 1
Order = TEL044, Comp = CAV-04, PSD = 91/07/03, PLS = 1

다음은 특정 Order의 각 Field내용을 List해 본 예제이다.

* Order number : TEL043

(Order Relation -----)
Order = TEL043, Comp = CAV-06, PSD = 91/06/26, PLS = 1

다음은 Planned Start Date순으로 Operation들을 List 해 본
것이다.

(Operation Relation -----)
Order = TEL041, Oper = 1, Mach = MIL01, PSD = 91/06/12
Order = TEL041, Oper = 2, Mach = DRL01, PSD = 91/06/14
Order = TEL041, Oper = 3, Mach = CEN01, PSD = 91/06/17
Order = TEL042, Oper = 1, Mach = MIL01, PSD = 91/06/19
Order = TEL041, Oper = 4, Mach = CMM01, PSD = 91/06/20
.
.
.
Order = TEL043, Oper = 7, Mach = ROB02, PSD = 91/07/08
Order = TEL043, Oper = 8, Mach = CMM01, PSD = 91/07/11
Order = TEL044, Oper = 5, Mach = EDM01, PSD = 91/07/11
Order = TEL044, Oper = 6, Mach = CMM01, PSD = 91/07/14
Order = TEL044, Oper = 7, Mach = ROB02, PSD = 91/07/15
Order = TEL044, Oper = 8, Mach = CMM01, PSD = 91/07/18

다음은 특정 Order 에 대한 Operation들을 List해 본 것이다.

* Order number : TEL041

(Operation Relation -----)

Order = TEL041.	Oper = 1.	Mach = MIL01.	PSD = 91/06/12
Order = TEL041.	Oper = 2.	Mach = DRL01.	PSD = 91/06/14
Order = TEL041.	Oper = 3.	Mach = CEN01.	PSD = 91/06/17
Order = TEL041.	Oper = 4.	Mach = CMM01.	PSD = 91/06/20
Order = TEL041.	Oper = 5.	Mach = GRN01.	PSD = 91/06/21
Order = TEL041.	Oper = 6.	Mach = ROB01.	PSD = 91/06/22
Order = TEL041.	Oper = 7.	Mach = CMM01.	PSD = 91/06/25

임의의 Record를 출력하고자 할때 존재하지 않는 Record이면 BDMR을 부른 Function에 Error Code를 반환해 주어서 그에 따른 적절한 대응을 BDMR을 사용한 Programmer가 할 수 있도록 하였다.