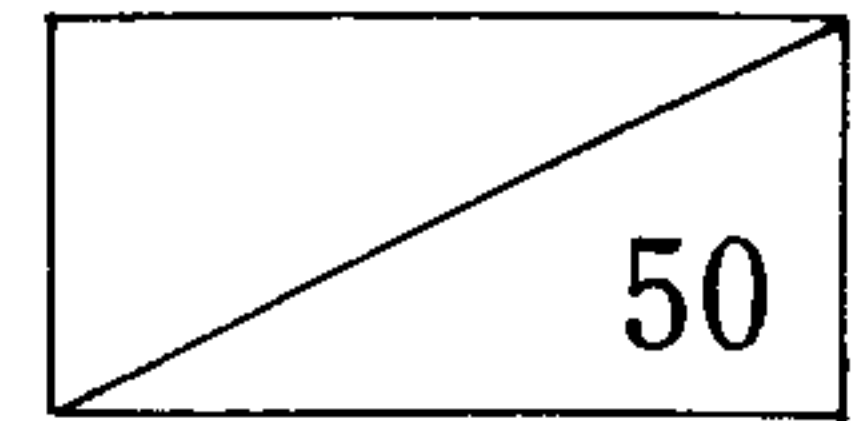


당 해 년 도  
최종보고서



자동화, 정보화 시범연구 및 기술지도 사업  
A project of Model Research and Technical Consulting  
on Integrated Manufacturing Information System

CIM 구축을 위한 CAD/CAM 통합  
운용에 관한 연구  
Integration of CAD/CAM Systems toward CIM

연구 기관  
한국과학기술연구원  
시스템공학연구소

과 학 기 술 처

# 제 출 문

과학기술처장관 귀하

본 보고서를 “자동화, 정보화 시범연구 및 기술지도 사업” 과제의  
(세부 과제 : CIM 구축을 위한 CAD/CAM 통합 운용에 관한 연구)  
최종 보고서로 제출합니다.

1993년 7월

주관연구기관명 : 한국과학기술연구원

시스템공학연구소

총괄연구책임자 : 박상봉(생산자동화연구단)

연구책임자 : 정연구(생산자동화연구단)

연구원 : 김현(생산자동화연구단)

심규창(생산자동화연구단)

여 백

# 요 약 문

## I. 연구 제목

CIM 구축을 위한 CAD/CAM 통합 운용에 관한 연구

## II. 연구의 목적 및 필요성

최근 설계와 가공에서 CAD/CAM 시스템의 도입이 매우 활성화되고 나아가 컴퓨터 통합 생산을 위한 요구가 높아져가고 있는 추세이다.

CAD/CAM은 CIM 구축을 위한 가장 중요한 기본 요소로서 표준화 촉진, 설계와 가공의 합리화, 설계/가공 일원화에 따른 생산성 향상 등 많은 효용성을 갖고 있지만 CAD/CAM 시스템의 도입만으로 이러한 결과를 내는 것이 아니라 이 시스템을 얼마나 효과적으로 운용하고 활용하느냐에 그 가치가 달려있으며 CAD/CAM이 CIM으로 발전되기 위해서는 전체 시스템과 유기적으로 통합될 수 있어야 한다.

그러나 국내 기업의 대부분이 자동화 기기나 각각의 단위요소의 부분적인 연계는 가능하지만 이들을 하나의 전체적 시스템으로 통합하기에는 아직도 많은 문제점을 갖고 있다.

특히 국내 도입되는 CAD/CAM 시스템이 stand-alone형이 주류를 이루고 있으며 매우 고가의 장비로써 주로 CAM을 지향하고 있다. 한편 기존의 제도판에서 행해지던 도면 작업은 저가의 PC용 2차원 CAD 시스템을 이용하고 있으며 이러한 시스템들은 상호 통합되어 있지 않고 또한 NC 가공 기계 등 자동화 기기와 긴밀한 통합을 이루지

못하여 설계와 가공의 일원화에 많은 어려움이 있다.

이에 본 연구에서는 전년도 금형 전용 설계 시스템과 중형 CAD/CAM 시스템을 통신 회선으로 연결하여 하나의 일원화된 시스템으로 관리 운용하고 NC 가공 기계와 연계하여 DNC 시스템을 구성함으로써 CIM의 가장 본질적인 문제 중의 하나인 통합에 그 목표를 둔다.

### III. 연구 내용

본 연구에서는 제품 설계에서 부터 가공에 이르는 과정을 CAD/CAM을 통하여 통합하고 CAD/CAM을 NC 가공 기계와 연계하며 이를 CIM의 기반 모듈로 확대시킨다.

전년도 금형 설계 전용 CAD 시스템을 이용하여 종래 제도판에서 행하던 도면 작업이 PC에서 파라메트릭 설계 방법에 의해 쉽게 행하여지며 간단한 2.5차원의 NC가공 데이터를 산출할 수 있다.

제품 설계를 위한 3차원 모델링이나 금형의 코어부에 대한 복잡한 형상 공구 위치 산출과 가공 시뮬레이션을 위해 중형 CAD/CAM 시스템을 활용하였고 이들 CAD 시스템과 CAD/CAM 시스템들은 LAN으로 연결되어 NFS를 통하여 데이터를 공유하며 상호 호환할 수 있도록 한다.

또한 CAD 설계로 부터의 기준 정보를 호스트 컴퓨터의 공통 데이터 베이스로 구축하고 이를 통합 운용 관리한다.

이에 따라 CAD/CAM 네트워킹이 이루어졌고 이종 CAD 시스템간의 데이터 호환을 위해 중간 파일의 형식이 연구되었으며 기하학적 형상 정보는 DXF나 IGES 형식으로 이종 CAD 시스템간에 호환할

수 있도록 기존 정보와 함께 CAD 소프트웨어를 이용하여 통합 운용 관리되었다.

이와 함께 셀 단위 자동화 단계에서 통합 생산 시스템을 이룩하기 위한 중요한 기본 요소로써 구축된 네트워킹 하에서 NC 공작 기계의 군관리에 의한 통합을 위해 DNC 시스템을 개발하였다.

이를 위해 CAD/CAM으로 부터 공구 위치 데이터를 기계에 맞도록 후처리 하는 포스트 프로세스를 개발하고 이를 RS232C 인터페이스를 통해 각 기계에 분배함으로써 기존의 테이프에 의한 가공을 벗어나 Tapeless와 온라인에 의한 NC 데이터의 공급으로 NC 가공 기계의 가동률을 향상시키고저 하였다.

전체 시스템은 C 언어를 이용 UNIX Openwindow 하에서 그래픽 유저 인터페이스로 프로그램되어 개발되었다.

본 연구는 설계에서 가공에 이르는 전 과정을 일원화시키기 위한 CAD/CAM의 통합 운용 시스템으로써 이를 통해 CIM의 Prototype을 제시하고자 하였다. CIM의 구성 요소를 살펴보고 이들을 통합하기 위한 기존 CIM의 모델을 검토하고 이를 참조로 CAD/CAM을 포함한 설계에서 생산에 이르는 기술 부문과 수주에서 출하에 이르는 관리 부문을 통합하기 위한 CIM Prototype을 개발 제시 하였다.

#### IV. 연구개발 결과 및 건의 사항

본 개발 시스템으로 CAD 설계의 성력화와 표준화 촉진에 기여하고 생산성을 높일 수 있을 것이다. 또한 셀 단위 자동화 단계에서 통합 생산 시스템을 구축하기 위한 DNC 시스템은 NC 가공 기계의 가동률을 향상시키고 현장 내의 기술 정보를 통합 운영함으로써



CIM의 기반을 구축할 수 있는 기본 모듈이 될 수 있을 것이다.

또한 본 시스템은 CIM을 위한 시스템으로써 이를 통해 CIM Prototype 개발 연구로 국내 제조업의 CIM 구축에 레퍼런스 모델을 제시하였다. 이는 CAD/CAM 시스템의 도입과 함께 CIM을 구축하기 위한 국내 기업에 지침이 될 수 있을 것이다.

본 시스템은 매우 기본적인 모듈들을 갖고 이를 확장 시킬 수 있도록 구성되었으나 실제 현장에 적용되지 않았기 때문에 이를 바로 범용적으로 이용하기는 어렵다고 생각된다.

따라서 본 연구가 한차원 더 발전 되기 위해서는 일차적으로 CIM의 각각의 구성 요소들에 대한 보다 더 고도화된 연구가 필요하며 2, 3차에 걸친 장기적인 연구 개발과 함께 국내 모든 시범 기업의 연구 참여와 이해가 필요하다.

아울러 기업의 연구 참여에 따른 자원 경영난을 지원할 수 있는 제도적 마련 역시 요구된다.

# SUMMARY

## I. Research Title

Integration of CAD/CAM Systems toward CIM

## II. Overview and Purpose

CAD/CAM technology is around twenty years old. The capability of CAD/CAM systems has evolved from simple two dimensional drafting systems to sophisticated integrated three dimensional design, drafting and engineering systems.

In general, CAD/CAM systems both improve the quality of a design and result in shorter product cycles because of increased lead time from a project's conception through the production of working drawings.

The development of a common database for project results in error reduction, better management control, and more feedback for the client. All this benefits result in more efficient drafting and design which result in cost savings for the firm.

Also electronics and computer technologies have had a significant influence on the control function in manufacturing. One important of this example of these technologies in production system is numerical control.

Unfortunately, the full potential has not yet been realized in the domestic industries at present. Although there have been continual efforts to construct integrated production systems, the actual current circumstances of the industries have not been stayed in island type topology and that is the current



stream of the technology in Korea.

Therefore, this project was tried to take into account such current situations of manufacturing firms.

The main objectives of the project were emphasized and focused on the integration of CAD/CAM systems and the interface of CAD/CAM systems and NC machines, which is meaningful start toward CIM system.

### **III. Result of Project**

CAD/CAM is one of the important components in CIM, but the value of it depends upon standardization, proper operation and total management.

This project was extended to a CIM foundation module by integrating CAD/CAM systems and interfacing CAD/CAM systems with NC machines, ranging from part design through scheduling to machining processing.

The suggested CAD system dedicated for molding in this project provides capability for parametric transformation for similar parts found in design stage and generating 2.5 dimensional NC process data.

For software development. C language was used. XView library was utilized for the development of user interface software on Openwindow environments and Sun Spark station system and commercially available CAD systems and CAD/CAM systems was utilized for design and accomplishing the cutting location data generation of complex geometric design. Every system is interconnected by LAN and designed to be shared data base system through NFS.

## **VI. Concluding Remarks**

In the past, Korea and several other countries have been successful in applying the penetration tactic. Which is to erode established industry leaders by penetrating a selected product category.

As time passes, conditions in foreign trade are changing and the competition is becoming more intense. The days of exporting modified existing products are coming to an end. Thus, the products which are exported must be products which are newly designed and developed.

There are many lessons to be learned from implementing CAD/CAM integration. It may expose numerous problems which may not have been considered as problems. The task will be hard and also rewarding.

Our current and future goals in addition to implementing CAD/CAM should be in the accumulation of design techniques and rapid prototyping to speed up the learning process.

여 백

# CONTENTS

Chapter I. Introduction .....	19
Section 1. Background .....	21
Section 2. Necessity of Project .....	25
Section 3. Purpose of Project .....	28
Section 4. Contents .....	30
Chapter II. Integration of CAD/CAM .....	33
Section 1. CAD Data Interface .....	35
Section 2. Development of Primary Data Management System .....	41
1. CAD/CAM Networking .....	41
2. Primary Data Generation .....	47
3. Primary Data Management System .....	51
Section 3. Development of DNC System .....	59
1. Interface issue in CAD and CAM .....	59
2. CL Data and Postprocessor .....	61
3. DNC System .....	68
Chapter III. Development of CIM Prototype .....	77
Section 1. Overview .....	79
Section 2. CIM Components and CIM Model .....	81
1. CIM Components .....	81
2. CIM Model .....	83

Section 3. Development of CIM Prototype . . . . .	86
Chapter IV. Conclusion . . . . .	91
Section 1. Concluding Remarks . . . . .	93
Section 2. Expected Effects . . . . .	98
Appendix . . . . .	101

# 목 차

제1장 서론	19
제1절 연구의 배경	21
제2절 연구의 필요성	25
제3절 연구의 목표	28
제4절 연구 내용 및 범위	30
제2장 CAD/CAM의 통합 관리 운영	33
제1절 이종 CAD 시스템 간의 데이터 호환	35
제2절 설계 기준 정보의 통합 관리 시스템 개발	41
1. CAD/CAM 네트 워킹	41
2. 설계로 부터 기준 정보 생성	47
3. 기준 정보 통합 운용 관리 시스템	51
제3절 DNC 시스템 개발	59
1. CAD와 CAM의 연계	59
2. CL Data 산출 및 포스트 프로세서 개발	61
3. DNC 시스템	68
제3장 CIM Prototype 연구	77
제1절 개요	79
제2절 CIM 구성 요소와 기존 모델	81
1. CIM 구성 요소	81



2. CIM 모델.....	83
제3절 CIM Prototype 개발 연구.....	86
제4장 결론.....	91
제1절 연구결과.....	93
제2절 기대성과.....	98
부록.....	101

# 그 립 목 차

그림 1-1. CAD/CAM 관점에서 본 CIM 발전단계.....	23
그림 1-2. 연구 개발의 목표.....	29
그림 1-3. 연구 내용 및 시스템 구성도.....	31
그림 2-1. 중간 파일 포맷을 이용한 데이터 교환방식의 예.....	36
그림 2-2. IGES의 기본 구조.....	38
그림 2-3. DXF 파일의 기본 구조.....	39
그림 2-4. OSI(Open System Interconnection)을 위한 ISO 모델.....	42
그림 2-5. 시스템 레퍼런스 모델.....	45
그림 2-6. CAD/CAM 네트워킹.....	46
그림 2-7. 설계 기준 정보 생성의 기본 구조.....	49
그림 2-8. 기준 정보 생성 모듈 메뉴 구성.....	50
그림 2-9. 기준 정보 생성 모듈 화면 예.....	51
그림 2-10. 기준 정보 운용 관리 모듈 구성도.....	52
그림 2-11. X-View 클래스 계층 구조.....	54
그림 2-12. 기준 정보 관리 화면 예(I).....	55
그림 2-13. 기준 정보 관리 화면 예(II).....	56
그림 2-14. 기준 정보 관리 화면 예(III).....	57
그림 2-15. 설계/가공의 일원화 프로세스.....	60
그림 2-16. CL(Cutter Location) 데이터.....	61
그림 2-17. CL 데이터와 CAMM-3 입력 데이터 비교.....	67
그림 2-18. CIM의 기술 부문의 기능적 구조.....	68

그림 2-19. DNC 시스템 구조도.....	70
그림 2-20. 컴퓨터와 NC간의 RS-232C핀 접속 예.....	73
그림 2-21. DNC 시스템의 화면 예.....	75
그림 3-1. CIM의 구성 요소.....	82
그림 3-2. NIST 5단계 계층 제어 CIM 모델.....	83
그림 3-3. CIM Prototype 구조도.....	86
그림 3-4. CIM 시스템 구성도.....	89
그림 3-5. CAD/CAM 시스템의 이용 형태.....	94
그림 3-6. 개발 시스템의 프로그램 구성도.....	96

## 표 목 차

표 1-1. CAD/CAM 도입의 필요성.....	26
표 2-1. NC 코드 어드레스.....	62
표 2-2. EIA-Standard에서의 G-Code.....	63
표 2-3. 포스트 프로세서 개발시 고려사항.....	66
표 2-4. EIA-RS-232C.....	72
표 2-5. 데이터 전송 방식.....	73

여 백

# 제 1 장 서 론



여 백

## 제1절 연구의 배경

최근 컴퓨터의 발달과 지난 30년 간의 공장 자동화의 급격한 진전에 따라 다품종 소량 생산에 대처하고 국내외 경쟁력의 강화를 위한 해결책으로 컴퓨터 통합 생산(Computer Integrated Manufacturing, CIM)의 실현이 절실히 요구되고 있다.

이러한 CIM은 오래전부터 논의 되어온 생산 현장의 자동화에 중점을 두고 있는 FA(Factory Automation)와는 달리 공작기계, 산업용 로봇 등의 개별 자동화 기계를 연결하고 수주에서 부터 개발, 설계, 생산 및 출하에 이르는 제반 생산 공정을 통합 통제하는 동시에 단위 기간 정보 및 물류의 통로를 형성하여 생산성을 향상 시키고자 하는 기술이다.

CIM의 구성 요소는 크게 생산 계획 및 통제 시스템(Production Planning & Control, PPC)을 통한 수주에서 출하까지의 관리부문과 CAD(Computer Aided Design), CAE(Computer Aided Engineering), CAPP(Computer Aided Process Planning), CAM(Computer Aided Manufacturing), CAQ(Computer Aided Quality Control)등 컴퓨터 지원을 통한 기술 부문으로 나누어 볼 수 있다.

또한 반도체, 컴퓨터 기술, 정보 전달 및 처리 기술등의 발전과 고객의 요구의 다양화에 따라 종래의 소품종 대량 생산 방식이 유연성을 갖는 다품종 소량 생산 방식으로 변화됨에 따라서 과거 Taylorism등에 따른 기능 분업화에서 벗어나 현재는 공통 데이터 베이스와 네트워크등

을 통해 자동화의 군도(Island of Automation)를 해결하기 위해 CIM에 의한 이들 구성 요소간의 통합을 위한 움직임이 활발해 지고 있는 실정이다.

따라서 CIM 내의 모든 구성 요소들은 궁극적으로 공통 데이터 베이스로써 통합된 정보의 흐름을 갖고 CIM 내에서 서로 유기적으로 연계되어야만 할 것이다.

CIM의 역사적 발전과정을 보면 생산 계획 및 통제 시스템을 통한 관리 부분의 자동화와 함께 CAD/CAM과 유연 생산 시스템(FMS)에 의해 생산 부분의 자동화 및 이들의 유기적인 통합화를 통하여 최근에는 지적 시스템인 Intelligent CIM으로 발전되고 있다.

그림 1-1은 CAD/CAM의 관점에서 본 CIM의 발전 과정이다. 1946년 최초의 컴퓨터가 만들어진 후 1954년 NC(Numerical Control)가 소개되었고 1956년 MIT의 Rose를 중심으로 APT(Automated Programmed Tool) 프로그램이 개발되어 CAM 시스템의 기초가 형성되었다. 1960년대에 이르러서는 NC의 사용과 함께 CAD, CAPP 및 CAM등이 대량 생산 시스템 하에서 생산성 향상에 각각 부문별로 독립적으로 활용되기 시작하였고 1970년대 접어들면서 CAD/CAM의 부분적인 통합화가 이루어졌으며 로봇 및 자동 창고 등 자동화 기기의 발전이 다품종 소량 생산에 대응하여 고안되었다.

1976년에는 ICAM(Integrated Computer Aided Manufacturing) 프로젝트가 이 미공군에 의해 수행되었고 1980년 중반에 CIM 프로그램으로 바뀌면서 CIM이 본격적으로 활성화 되기 시작하였으며 현재 지적 생산 시스템의 연구와 더불어 Intelligent CIM으로 발전되고 있다.

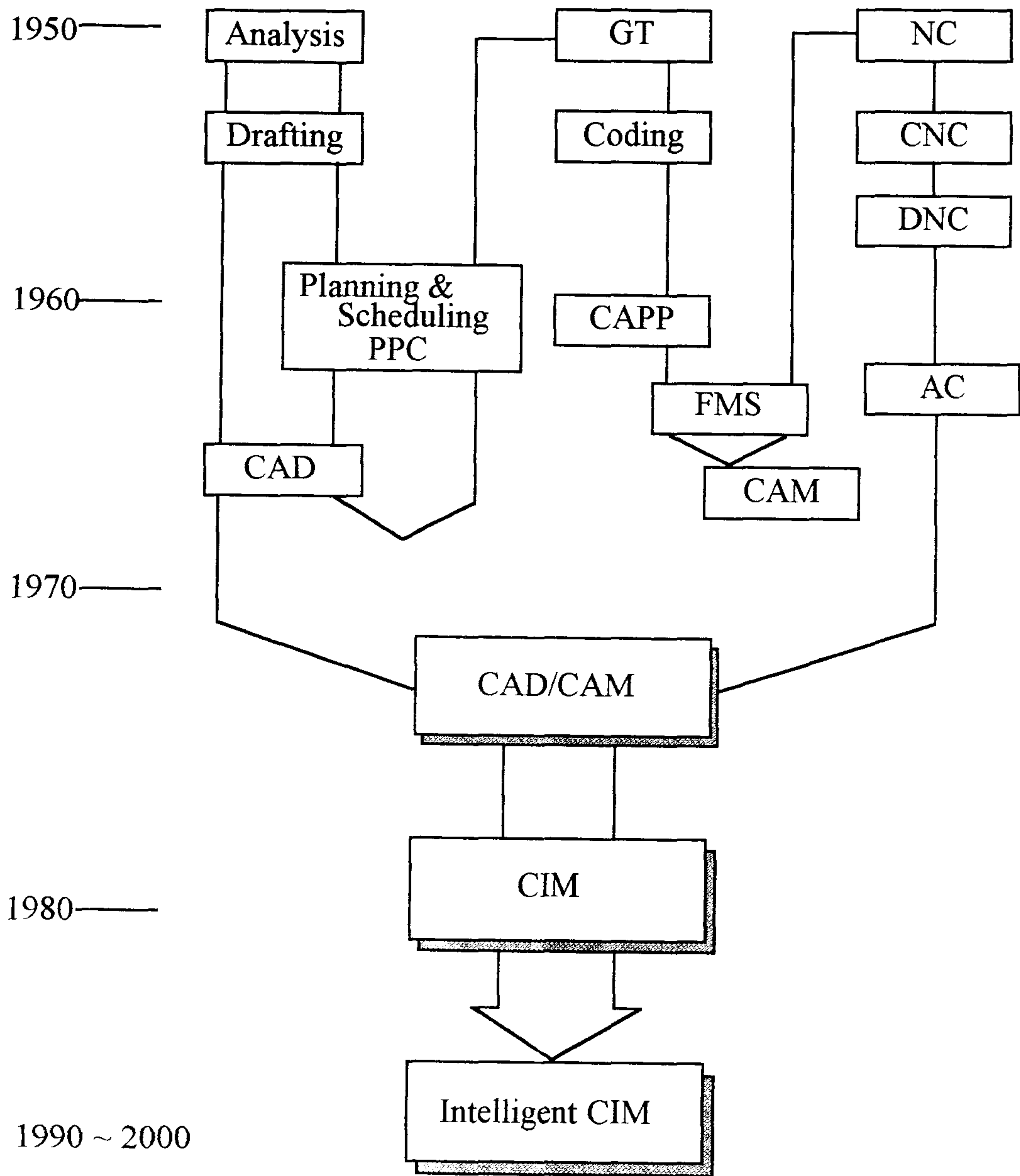


그림 1-1. CAD/CAM의 관점에서 본 CIM의 발전 단계

이러한 배경속에서 CAD/CAM은 CIM의 매우 중요한 기본 단계로서 현재 국내에서도 이미 많은 기업들이 그 중요성을 인식하고 설계 및 가공 분야에서 CAD/CAM을 도입, 이용하고 있으며 이는 설계, 제조, 생산, 관리등의 생산에 직접 관계되는 부문에서 영업까지를 컴퓨터에 의해 통합되고 수주에서 출하까지의 활동을 효율적으로 수행함으로써 납기 단축이나 비용절감, 품질 향상 등으로 기술적 비교 우위에 서려는 노력과 함께 CIM으로 발전 시키기 위해 노력하고 있다.

특히 국내 제조 업체에서 CAD/CAM 시스템을 도입하여 어떻게 활용하느냐에 따라 설계 및 가공의 합리화, 공수 저감 및 납기 단축에 크게 기여할 수 있고 또한 금전적 가치에 우선하여 Know-How의 축적과 체질 개선의 측면에서도 높은 효과를 기대할 수가 있으므로 CAD/CAM 시스템은 도입과 함께 그 활용에의 관심도가 매우 커져가고 있다.

그러나 최근 컴퓨터 환경의 변화와 함께 국내 도입되는 대부분의 CAD/CAM 시스템은 Stsnd-alone형이 주류를 이루고 있으며 매우 고가의 장비로서 설계 및 제도를 위해 이용하기 보다는 가공을 위한 장비로 주로 CAM 시스템을 지향하고 있어 보통의 도면 작업은 종래와 같이 제도판 위에서 행해지거나 저가형 CAD 시스템을 이용하고 있다.

또한 이들 시스템들은 NC 공작 기계와 같은 자동화 기기나 CAD/CAM 시스템을 포함한 CIM 내의 각 구성 요소들 간의 부분적인 연계는 가능하지만 이들을 하나의 전체적인 시스템적 흐름을 형성하기에는 아직도 많은 문제점을 갖고 있어 설계와 가공의 일원화 및 통합 생산 시스템 구축에 많은 어려움이 있다.



## 제2절 연구의 필요성

최근 경제 사회의 고도 성장과 함께 소비자의 요구가 다양해지고 이로 인한 제품의 다양화와 제품의 라이프 사이클이 단축됨에 따라 과거 대량 생산 체제가 시장 중심의 다품종 소량 생산 체제로 바뀌게 되었다.

이와 같이 생산 체계가 변모함에 따라 설계 및 가공 부문에서도 성력화, 합리화 및 표준화를 이루는 획기적인 생산성 향상의 요구로 CAD/CAM 시스템이 도입되고 있다.

실제 CAD/CAM 시스템의 도입에 대한 필요성은 표 1-1과 같이 생각할 수 있으며 이러한 CAD/CAM 시스템은 그 필요성에 직접적으로 기여하는 효과와 함께 금전적으로 환산할 수 없는 효과가 있음을 고려할 때 이러한 필요성에 부응해 앞으로도 도입이 크게 증가될 것으로 고려된다.

그러나 이러한 CAD/CAM 시스템이 매우 활성화 되고 현재 국내 제조업에서 설계 및 가공의 합리화, 공수 저감 및 납기 단축에 그 효용성을 인정 받고 있지만 실제 이러한 CAD/CAM 시스템을 도입하였다 고 하더라도 그 즉시 효과를 내는 것이 아니라 이 시스템을 얼마나 효과적으로 잘 활용하고 어떻게 전체 시스템과 함께 유기적으로 통합 일원화 할 수 있느냐에 따라 그 가치가 결정된다고 할 수 있다.

또한 CAD/CAM 시스템은 CIM에서의 가장 중요한 기본 요소 중의 하나로써 CAD/CAM 시스템의 도입과 함께 기업 내의 CIM 구축에 발판이 마련될 수 있을 것이다.



표 1-1. CAD/CAM 시스템 도입의 필요성

환경 변화	CAD/CAM 도입 필요성
시장 환경의 변화	<ul style="list-style-type: none"> <li>- 소비자 요구의 다양화</li> <li>- 가격 경쟁의 격화</li> <li>- 국제 경쟁의 격화</li> <li>- 제품 지식의 집약화</li> <li>- 제품 Life Cycle의 단축</li> </ul>
설계 환경의 변화	<ul style="list-style-type: none"> <li>- 신제품 개발 경쟁의 격화</li> <li>- 고품질, 저가격화의 설계 필요성 증대</li> <li>- 설계 납기의 단축</li> <li>- 제품 사양의 다변화로 설계 작업량 증대</li> </ul>
제조 환경의 변화	<ul style="list-style-type: none"> <li>- 다품종 소량 생산</li> <li>- 생산 자동화 추세</li> <li>- 설비 기계의 가동률 향상의 지향</li> </ul>
인적 환경의 변화	<ul style="list-style-type: none"> <li>- 고학력화 및 인력 고령화에 의한 업무 자동화 요구 증대</li> <li>- 규정 근무 시간 유지 요구</li> <li>- 단순 반복 작업의 생산성 저하를 창조적 작업으로써의 지향</li> <li>- 숙련된 기능 인력의 부족</li> </ul>

이에 전 년도에 형 구조 및 부품 표준화에 의한 금형 설계 전용 CAD 시스템 개발로 설계의 효율화를 높였고 이를 가공과 연계하기 위해 CAD 정보로부터 WireEDM 가공기의 NC 가공 데이터를 직접 산출함으로써 설계 및 가공을 일원화 시키고자 하였다.

이는 설계 및 도면 작성과 2차원 가공 데이터 산출의 측면 등 단위 부분에서는 매우 효율적이거나 이러한 CAD/CAM 시스템을 실제 운용하는 부분과 CAD/CAM 시스템과 생산 관리 부분과의 통합 차원에서는 해결해야 할 부분이 있는게 사실이다.

실제 금형 업계등의 국내 제조업 부문에서, 아직은 CAD/CAM 시스템이 고가의 장비로써 설계나 도면 작성등은 아직도 제도판을 이용하거나 저가의 PC용 CAD 시스템을 이용하는게 일반적이고 이러한 고가의 CAD/CAM 시스템은 대부분 NC 가공 데이터를 산출하기 위한 즉 가공을 위한 CAM 장비로써 이용되고 있다.

또한 국내 제조업체에서 이러한 CAD/CAM 시스템은 자체로써 그 이용도가 과거에 비해 매우 높아졌으나 NC 공작기계 등의 자동화 기계와 긴밀한 통합을 이루지 못하는 등 전체시스템에서 볼 때 역시 균도를 이루며 설계/가공의 일원화 및 통합 생산 시스템의 구축에는 많은 어려움이 있는 실정이다.

이에 따라 여러 종류의 CAD 시스템들에 대한 도면 데이터의 관리, 설계에서 산출되는 기준 정보 데이터 관리, CAD와 CAM의 원활한 연계, CAD/CAM과 NC 가공 기계와의 연계등 CAD/CAM 시스템의 운용 및 CIM 구축을 위한 CAD/CAM 부문에서의 통합에 대한 요구가 절실해져 가고 있다.

### 제3절 연구의 목표

이에 본 연구에서는 CAD 시스템 간의 도면 및 기준 정보 데이터의 통합, CAD와 CAM간의 연계, CAD/CAM 시스템과 NC가공 기계와의 연계 등을 통하여 설계/가공의 하나의 통합된 시스템을 구축하고 장기적으로는 통합 생산 방식인 CIM을 이루기 위한 발판을 구축하고자 하는 것이 본 연구의 목표이다.

1차년도에서 개발된 금형 설계/가공용 CAD 시스템을 보완하고 이를 중형 CAD/CAM 시스템과 연결하여 상호 연계를 위한 방법을 모색하며 이들 CAD/CAM으로 부터의 데이터를 RS-232C를 이용, NC 공작기계와의 통합을 꾀함으로써 CIM의 가장 본질적인 *Integration*에 그 목표를 둔다.

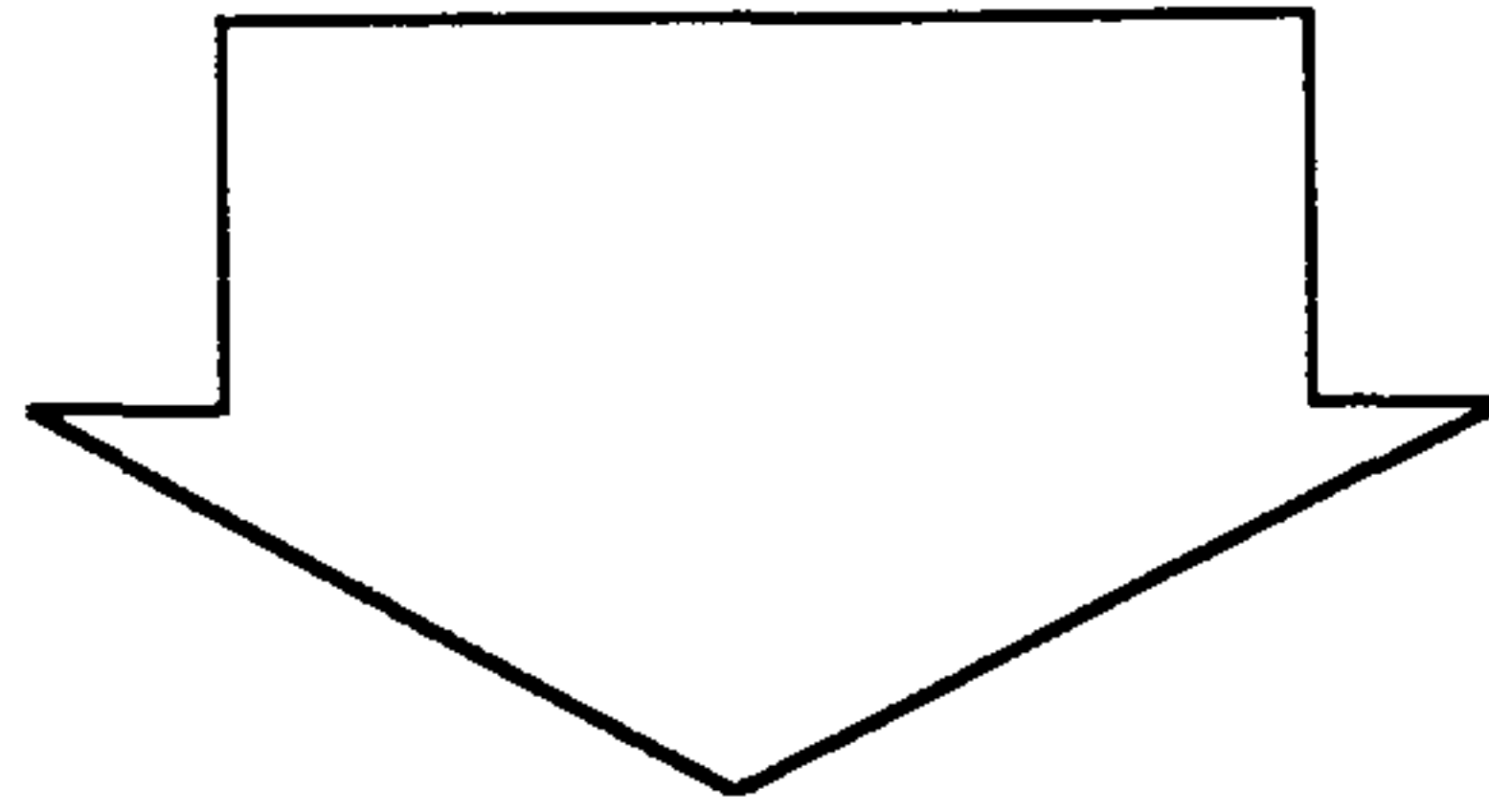
이를 위해 기본적으로 CAD 시스템을 위한 네트워크에 대한 고찰과 CAD 시스템간의 데이터 호환을 위한 DXF 파일 분석 작업이 이루어지고, CAD/CAM 시스템으로 부터의 공구 위치 데이터(Cutter Location Data)에 대한 포스트 프로세스 작업이 수행되었다.

또한 이러한 가공 데이터의 관리와 NC 가공 기계와의 직접적 연계를 위해 계층적, 분산적 방식의 DNC(Distributed Numerical Control) 시스템으로의 확장이 연구되어, 이를 통해 CAD/CAM 내에서의 통합된 CIM Prototype을 제시하고자 하였다.

본 연구는 본 연구단에서 현재 진행 중인 한국형 통합 공장 관리 시스템의 일정 계획 및 통제 시스템과 통합되어 생산 시스템을 이루기 위한 중요한 기본 모듈이 될 것이다.

본 연구의 세부 목표는 그림 1-2와 같다.

CAD/CAM 통합 운용
-이종 CAD/CAM 시스템 간의 네트워크
-CAD 데이터 관리 및 운용 시스템
-NC 가공 데이터 관리 및 통제 시스템
-계층적 분산 DNC 시스템



CIM Prototype 연구
-전체적으로 통합된 CAD/CAM Flow 구성
-CIM Prototype 개발 연구

그림 1-2. 연구 개발의 목표

## 제4절 연구 내용 및 범위

금형 업계를 포함한 국내 제조업체에서 CAD/CAM 시스템의 도입과 함께 가장 큰 문제로 부각되는 것이 설계에 있어서 표준화에 의한 각 기업에 요구되는 전용 설계 시스템 구축과 함께 CAD/CAM 시스템 및 자동화 기기와의 통합을 통한 설계/가공의 일원화에 있다고 할 수 있다.

실제 기업에서 설계 작업을 위해 기존의 제도판을 없애고 CAD 시스템으로 대체 하고자 할 때 설계/가공의 일원화를 고려하기 위하여 고가의 중형 CAD/CAM 시스템을 전부 설치한다는 것은 경제성에 미루어 어려운 일일 수 밖에 없고 보통은 도면 작성을 위한 저가의 PC 용 CAD 시스템이 제도판을 대체하여 이용되며 설계 해석이나 가공 데이터 산출 등 각 설계 작업의 특성에 맞는 여러 종류의 시스템들이 혼용되어 사용되는 추세에 있다.

따라서 표준화 등에 의한 자사에 맞는 CAD/CAM 시스템의 활용과 함께 제품 설계에서 부터 가공에 이르는 과정을 컴퓨터로 통합하기 위한 시스템을 구축하기 위해서 각 구성 요소간의 연계와 관리 운용 및 통제가 절실히 요구된다.

이에 본 연구에서는 그림 1-3과 같은 통합된 시스템적 흐름을 형성하여 이들의 유기적인 상호 연계와 관리, 운용하기 위해 표준 CAD 데이터 포맷의 하나인 DXF(Drawing Interchange Files)에 의한 도면 데이터 및 설계에서의 기준 정보 데이터를 공통 데이터 베이스로 구축하고 이를 관리, 운용하기 위한 시스템을 구축 하였다.



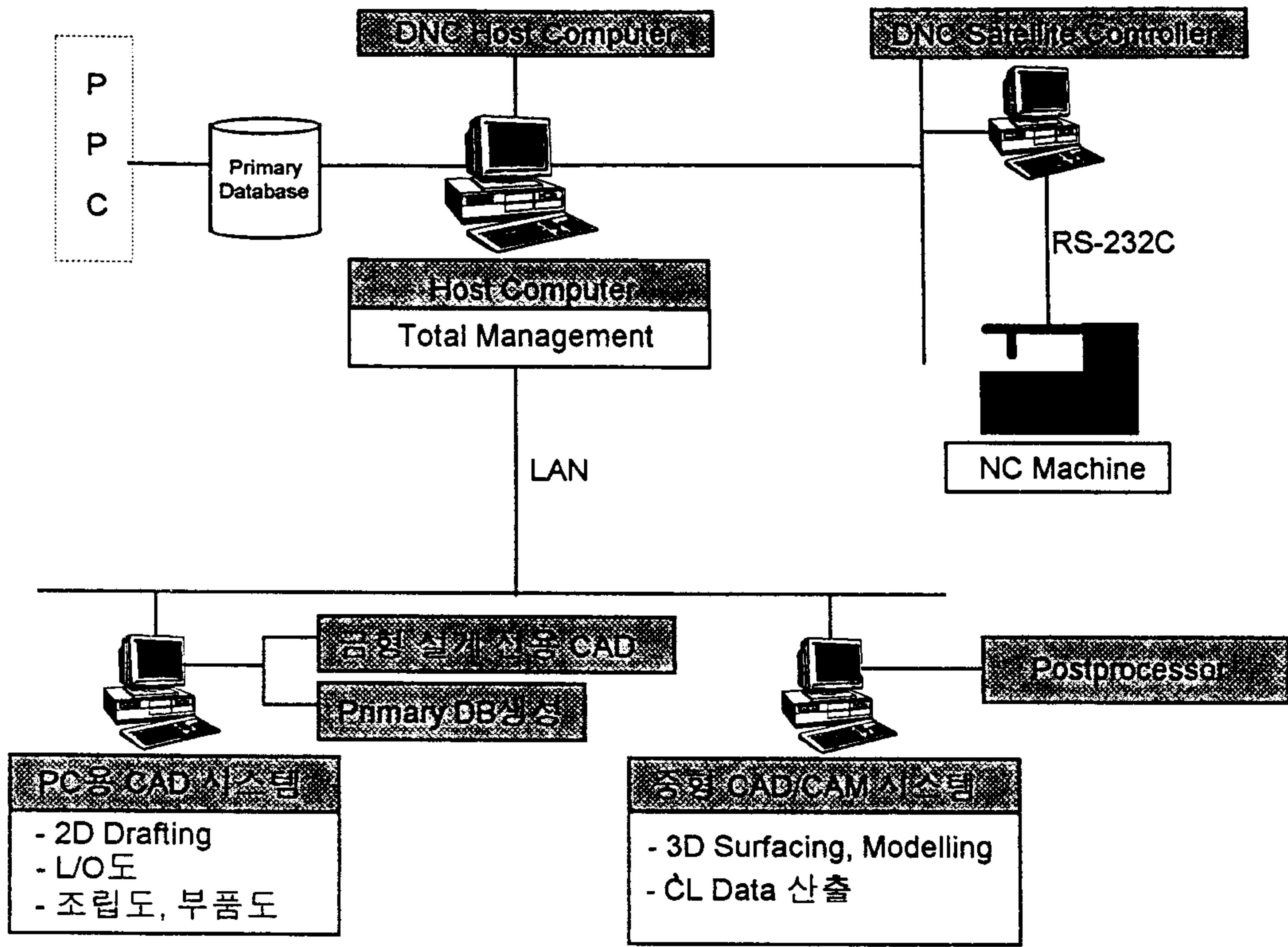


그림 1-3. 연구 내용 및 시스템 구성도

구조 및 부품의 설계는 전 년도 금형 설계 전용 CAD 시스템을 이용하여 종래 제도관 위에서 행해지던 도면 작업이 PC에서 행해지며 간단한 2.5차원 가공 데이터가 산출 될 수 있다.

이와 함께 제품 설계를 위한 3차원 모델링이나 금형 코어부의 형상 가공 등을 위해 중형 CAD/CAM 시스템이 활용되었고 이로 부터 얻어지는 CL 데이터는 실제 가공할 NC 가공 기계의 특성에 맞도록 포스트 프로세스 작업이 행해지므로써 NC 가공 데이터가 산출된다.



또한 이러한 NC 가공 데이터는 PPC측의 생산 일정 계획으로 부터의 작업 일정 계획에 의하여 NC 가공에 필요한 데이터를 각 NC 가공 기계에 Download 되어 가공할 수 있는 DNC 시스템이 개발된다.

이들 전체 시스템은 EtherNET을 통한 LAN(Local Area Network)으로 연결되었고 컴퓨터와 NC 가공 기계와의 연계는 RS-232C 인터페이스가 이용되었다. 또한 이들 각 구성 요소들로 부터의 데이터는 기준 정보 데이터와 함께 호스트 컴퓨터에 공통 데이터 베이스로 구축되었으며 이를 관리할 수 있는 모듈이 호스트측에서 개발되었다.

이 시스템은 CAD/CAM을 중심으로 하여 CIM의 기본 구성 요소로서의 Prototype으로 제시되고 실제 현장에서 활용될 수 있도록 하였다.

## 제 2 장 CAD/CAM의 통합 관리 운용

여 백

## 제 2 장 CAD/CAM의 통합 관리 운용

### 제1절 이종 CAD 시스템 간의 데이터 호환

1980년 경에는 CAD 시스템이 고가의 장비로 명확한 효과를 평가할 수 없었던 이유등으로 일부 대기업에서만 활용되었으나 EWS의 등장과 PC CAD 시스템의 기능 확장 및 저가격화에 힘입어 CAD/CAM 시스템의 보급이 크게 확산 되었다.

이에 따라 각 기업 내에 여러가지 목적의 다양한 CAD 시스템이 가동되고 있고, 또한 관련 기업이 서로 다른 CAD 시스템이 가동되고 있는 실정이다.

CAD 데이터의 호환은 개발과 동시에 대두되어 온 문제로써 1970년대 이후로 그 중요성이 인식되어 많은 연구가 행해져 왔으나 현재는 일반적으로 CAD 시스템간의 인터페이스를 통해 데이터를 직접 변환하기 보다는 중간 파일 포맷을 사용하여 데이터를 교환한다.

A 시스템에서 B 시스템으로 직접 데이터를 넘겨주는 인터페이스 프로그램을 개발할 경우 충실한 변환을 실현할 수 있어 실행 효율이 좋다는 장점이 있으나 이는 시스템의 확장 때마다 수정하지 않으면 안되므로 이상적이지 않고 중간 파일의 변환은 시스템 확장이나 CAD 표준 개정 여부에 직접적인 영향을 받지 않고 CAD 시스템 상호간에 자유로운 데이터 호환이 가능하나 중간 파일을 통한 데이터 교환 자체가 현재로써는 쉽지 않고 실행 시간이 길다는 단점이 있다.

이들 중간 파일 포맷은 미국 규격 ANSI를 획득한 IGES(Initial Gra-

phic Exchange Specification)와 AutoDesk사에서 개발되어 PC용 CAD간의 파일의 표준이 되고 있는 DXF (Drawing Exchange Files), 독일에서 개발된 VDA-FS나 프랑스의 SET등이 주로 이용된다.

국내에 도입 운용되는 CAD 시스템의 중간 파일 포맷으로의 변환이 2차원의 경우 DXF가 매우 대중화되어 있고 중형 CAD/CAM 시스템의 경우 IGES가 기본적으로 표준포맷이 되어 있다.

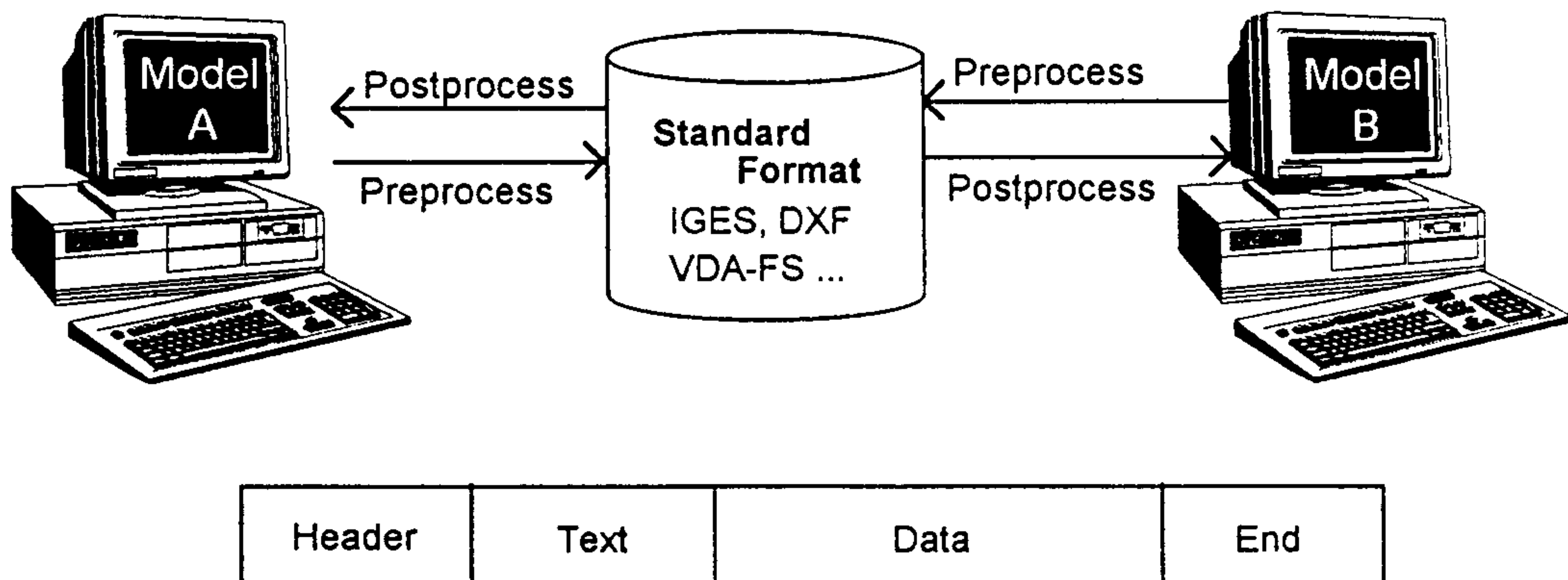


그림 2-1. 중간 파일 포맷을 사용한 데이터 교환 방식의 예

일반적으로 PC용 CAD 시스템의 경우는 DXF 파일을 직접 입출력할 수 있으나 MicroCADAM등과 같이 DXF를 직접 받아들이지 못하는 경우에는 변환기를 통해 MicroCADAM의 중간 파일인 BMI로 변환한 뒤 데이터를 건네준다.

그러나 CAD 소프트웨어의 데이터 구조는 도형 데이터와 속성 정보, 문자 정보, 면 정보, 솔리드 정보 등의 물체와 그 속성 정보, 곡면

정보뿐만 아니라 Layer, View, User Coordinate System, Block 등 기타의 많은 정보를 갖고 있으며 현재의 고기능인 CAD 데이터를 낮은 레벨인 중간 파일 형식으로 변환하기에는 지원되지 않는 요소가 많으며 각 메이커의 해석 조차 통일되지 않고 있어 서로 다른 CAD 소프트웨어 사이에서 정보를 완벽하게 호환 하기는 어려운 실정이다.

최근 복잡한 CAD 정보의 표준 형식의 변환을 위해 NIST(National Institute of Standards and Technology)의 지도하에서 PDES(Product Data Exchange Specification) 계획이 진행중에 있으며 ANSI와 ISO를 통해 개선된 국제 표준으로 제출되었다. 이 PDES는 기존 IGES의 개선으로써 IGES의 기능을 완전히 통합하여 발전될 것이다. 하지만 현재로서는 IGES나 DXF 표준 형식이 많이 쓰이고 있는 실정이다.

IGES 개발의 시초는 1979년 미국 표준국(NSB), 보잉, General Electric 등에 의해 CAD/CAM 시스템 사이의 데이터 베이스 변환을 위한 중간 포맷 개발을 위한 프로젝트의 추진이었고 1980년 1월 IGES가 제정되었으며 1981년 9월 미국 ANSI(Y14.26) 규격에 의해 채택되었다. 그 후 미국의 주요 CAD 벤더들 사이에 지원을 받아 현재는 도형 데이터 변환을 위한 가장 대중적인 규격으로 자리잡았다.

IGES에 의한 데이터는 보통 수천에서 수만에 이르는 카드 이미지로 구성된다. IGES 규격에서는 그것을 4개의 그룹으로 나누어 각각을 Start Section(S), Global Section(G), Directory Section(D), Parameter Section(P) 및 Terminate Section(T)으로 구별하였다. 그림 2-2는 각 Section의 보기를 나타낸다.



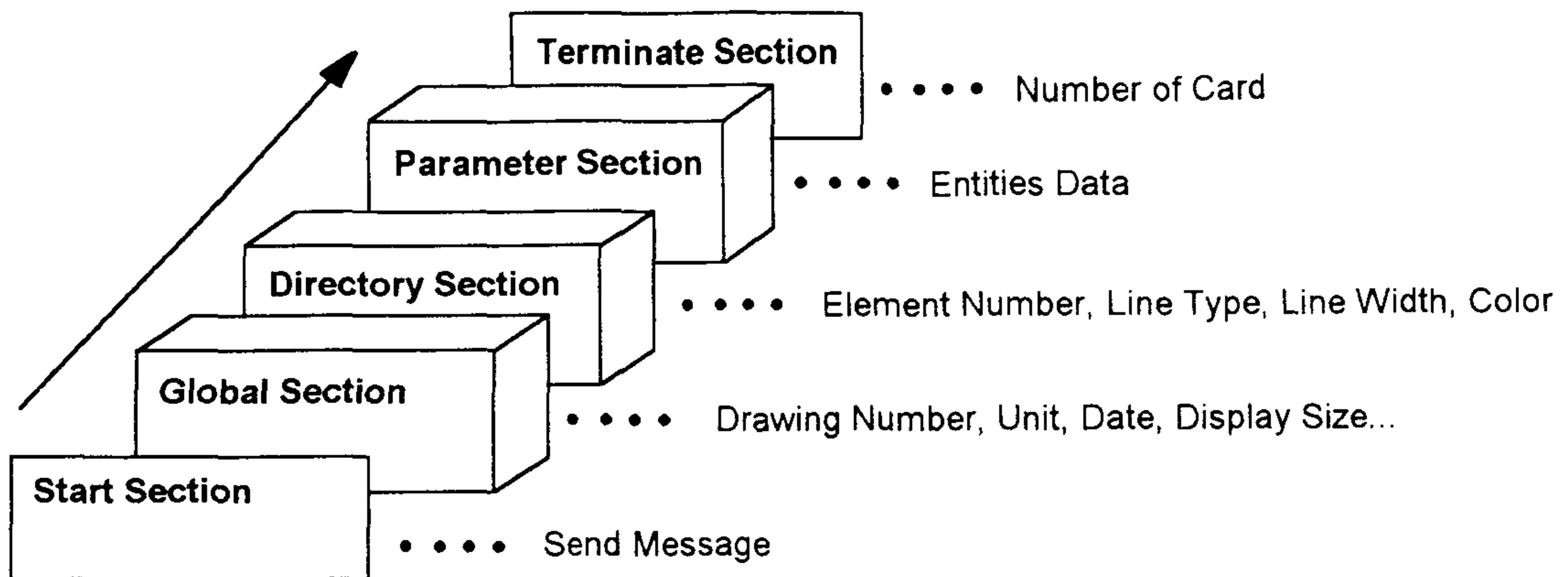


그림 2-2. IGES의 기본 구조

DXF 파일은 원래 AutoDESK사가 개발, 보급한 것이지만 국내에서 AutoCAD의 보급이 매우 대중화 되면서 현재는 2차원 PC용 CAD의 중간 파일의 표준과 같은 위치를 점유하고 있는 실정에 있다.

DXF 파일은 크게 HEADER 부분, TABLES 부분, BLOCK 부분, ENTITIES 부분과 END OF FILE로 구성되는 ASCII TEXT FILE이다. 다음 그림 2-3은 Section 표시와 Header로만 구성된 빈 DXF 파일의 보기이다.

일반 범용 CAD 시스템인 AutoDESK사의 AutoCAD와 Computer Vision사의 Personal Designer간에 DXF를 통해 데이터를 호환 할 경우 문자 (Text)를 제외한 선, 원등의 기본 Drawing Entity의 경우는 거의 완전한 호환을 이루고 Polyline을 포함하여 타원이나 다각형 등은 Polyline 형태로 변환되어 호환이 가능하였다.



```

0          (Begin HEADER Sec.)
SECTION
2
HEADER
    《Header variable items go here》
0
ENDSEC          (End HEADER Sec.)
0          (Begin TABLES Sec.)
SECTION
2
TABLES
0
TABLE
2
VPORT
70
    《Viewport table items go here》
0
ENDTAB
0
TABLE
2
LTYPE, LAYER, STYLE, VIEW, USE or DRGMGR
70
    《Table items go here》
0
ENDTAB
0
ENDSEC          (End TABLES Sec.)
0          (Begin BLOCKS Sec.)
SECTION
2
BLOCKS
    《Block definition entities go here》
0
ENDSEC          (End BLOCKS Sec.)
0          (Begin ENTITIES Sec.)
SECTION
2
ENTITIES
    《Drawing Entities go here》
0
ENDSEC          (End ENTITIES Sec.)
0
EOF          (End of File)

```

그림 2-3. DXF파일의 기본 구조

3차원의 경우 3D Surface는 그대로 호환되며 Ruled Surface 또는 Revolution Surface등의 Surface Model의 경우는 Polyline과 Mesh형태로 변환되어 서로 호환이 가능함을 알 수 있었다.

그러나 3차원 모델링 데이터의 경우 AutoCAD는 CGS방식에 의한 Solid Model을, Personal Designer는 Ruled Surface나 Sweep 등의 방법에 의한 Surface Model을 지원하고 있기 때문에 근본적으로 이들의 상호 호환은 힘든 것을 알 수 있었다.

이에 본 연구에서는 PC용 CAD 시스템 간의 데이터 호환은 DXF로 실행하고 중형 CAD/CAM 시스템의 경우는 3차원 모델에서 IGES 프로세스를 이용하였다.

또한 CAD 시스템 간의 DXF 포맷으로 변환 시 이종 CAD 시스템의 경우는 상호 호환성을 높이기 위해 가능한 Layer, View, User Coordinate System, Block등을 제외한 Entities Section만을 변환하여 Drawing 부분만을 호환하도록 하였다.

설계 도면에 대한 CAD 파일이 DXF 포맷으로 공통 데이터 베이스에 저장되고 각 이종 CAD 시스템에서 필요시 각각의 포스트 프로세스를 거쳐 도면 데이터를 받아 들인다.

한편 부품의 기하학적 표현은 생산에서 보다 많은 활용을 할 수 있는 정보를 포함하고 있고 가공과의 연계 시에 반드시 필요하다.

가공 데이터를 산출하기 위한 CAM 시스템이 CAD 시스템으로 부터의 기하학적 정보를 활용하기 위해서도 중간 파일 형태는 반드시 필요하며 이 때는 3차원 형상 모델링이 요구되고 이러한 3차원 모델에 대한 데이터 호환은 DXF에 비해 IGES가 범용성이 크고 호환성이 높기 때문에 이를 활용한다.

## 제2절 설계 기준 정보의 통합 관리

### 2.1. CAD/CAM 네트워킹

앞서 기술한 바와 같이 PC CAD 시스템의 사용이 늘고 중형 CAD/CAM 시스템과의 연계가 요구되면서 CAD/CAM 네트워킹은 일반 시스템 이용자에게 많은 문제점으로 부각되고 있다.

특히 CIM 구축 시 컴퓨터에 의한 통합을 위해 각 부분 간의 네트워킹은 매우 중요하며 CAD/CAM 시스템의 통합 운용을 위해서도 반드시 필요하다.

또한 인터 네트워킹(Interntworking)의 빈도수가 높아짐에 따라 전송 데이터의 양이 많아지고, 파일의 크기도 대형화되어 파일 서버의 성능이 심각한 문제로 대두되고 있어 이를 위한 전용 서버의 도입이 불가피하게 되었다.

이러한 파일 서버는 공통의 데이터 베이스를 저장하게 되고 이는 모든 부문에서 액세스되어 활용될 수 있도록 구축되어야만 한다.

네트워크의 요구는 LAN(Local Area Network)의 등장과 함께 데이터 공유 및 일원 관리를 위해 급격히 발전 되어 갔다.

LAN은 다수의 독립적인 컴퓨터 기기들이 상호 간에 통신이 가능하도록 하는 데이터 근거리 통신망을 말하고 분산된 데이터를 처리하고 자원을 공유할 수 있으며 중요한 데이터에 대한 제한된 액세스로 보안을 유지할 수 있다.

OSI(Open Sysyem Interconnection)에서는 통신 프로토콜(통신규약, Protocol)을 맞추기 위해 프로토콜 레이어를 7 레이어로 참조 모델을 규정

하였고 이를 ISO/OSI 7 Layer Model이라고 하며 LAN 프로토콜은 ISO/OSI 7 Layer의 하위층(물리층, 데이터 링크층)에 위치한다.

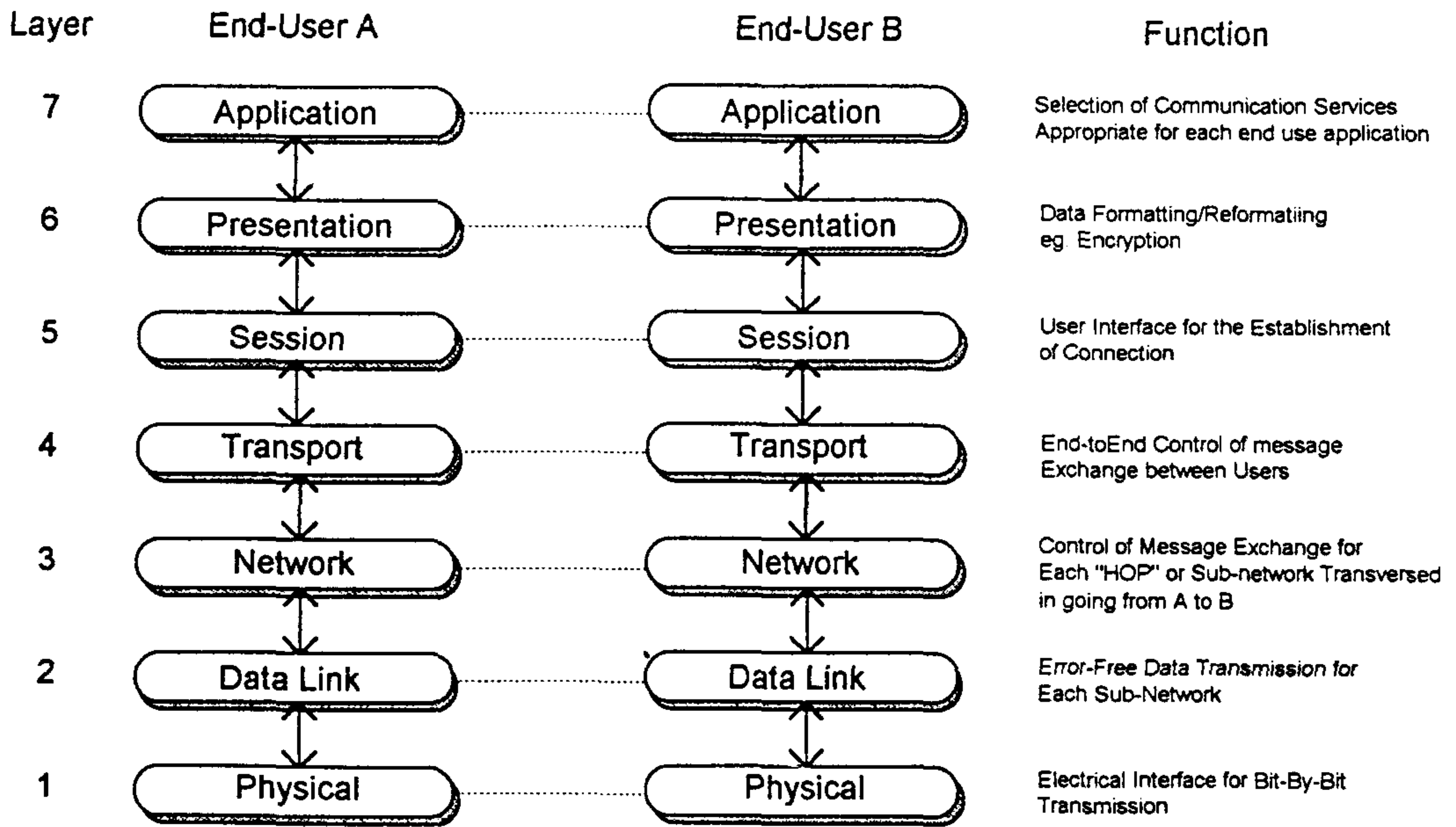


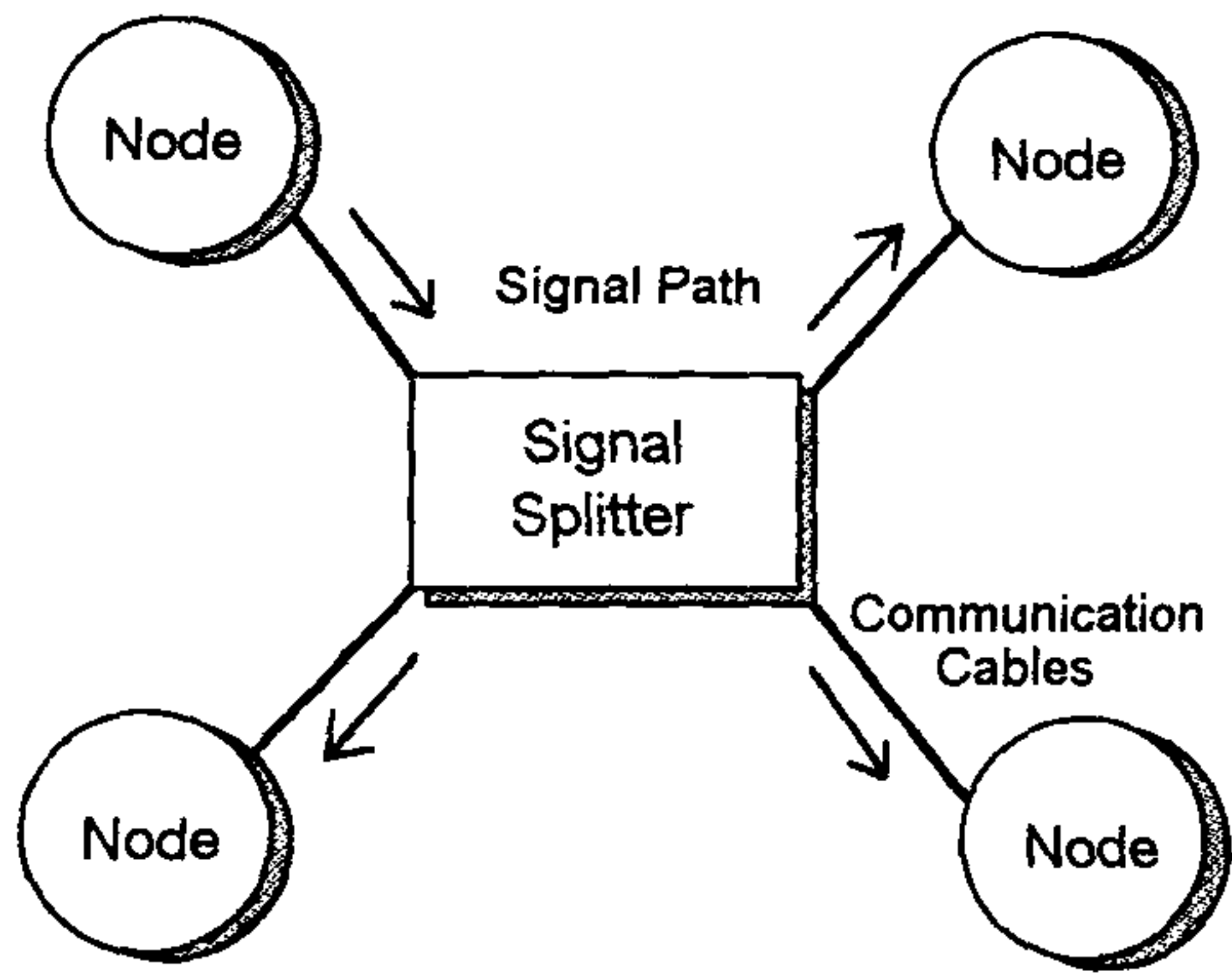
그림 2-4. Open System Interconnection을 위한 ISO 모델

보통 LAN의 기본적인 형태(Topology)는 다음과 같이 Star Bus형태, Ring형태, Star Ring 형태 및 Linear Bus 형태로 나누어 진다.

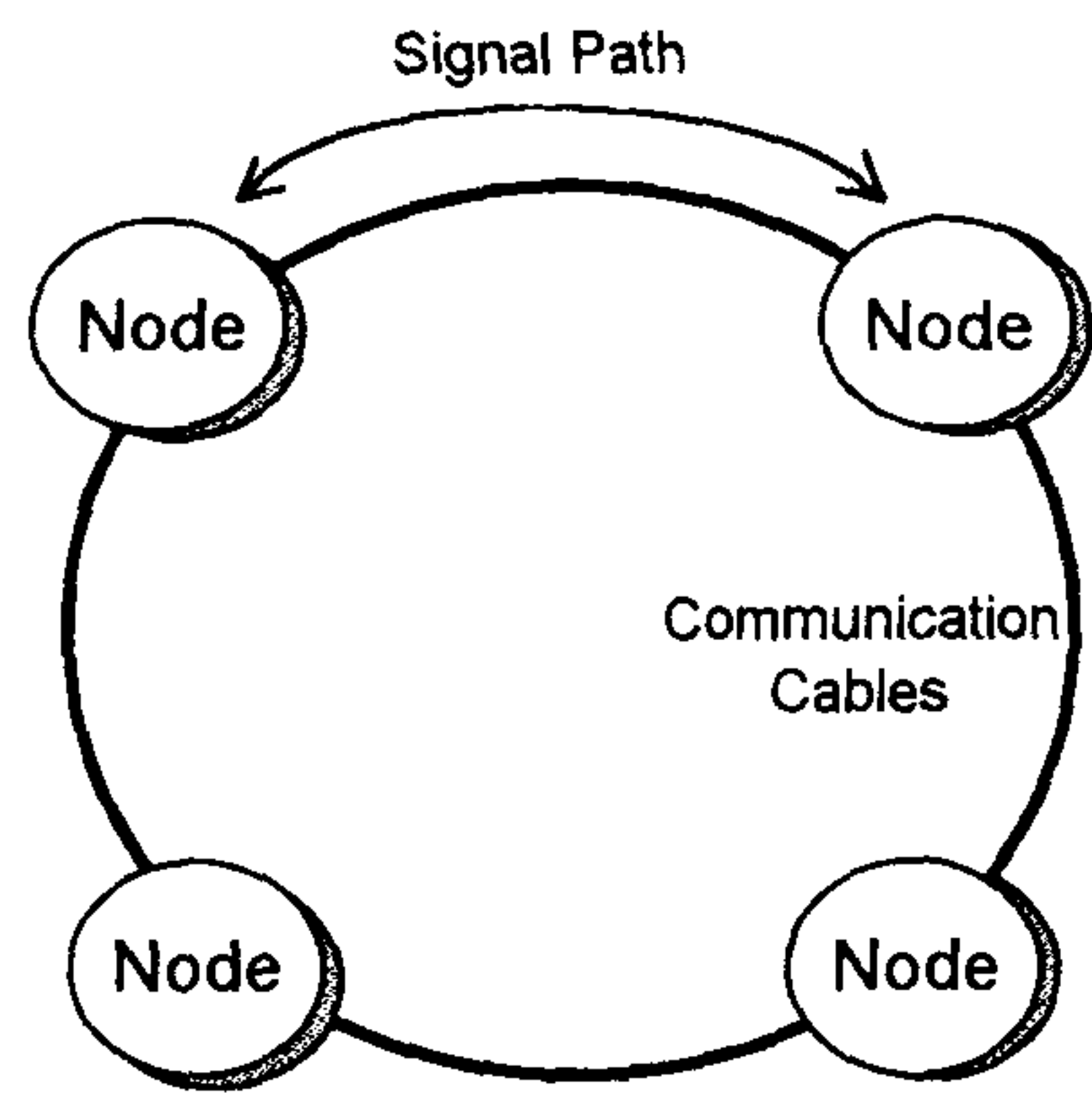
정보 시스템 구축을 위한 네트워크는 ISO를 포함한 많은 부문에서 그 표준화를 위해 꾸준히 노력 해오고 있으며 현재 공장 자동화를 위한 생산 네트워킹을 위해 MAP(Manufacturing Automation Protocol)이 크게 부각되어 논의되고 있다. 이러한 MAP은 전체 시스템의 LAN중 기계 레벨과 기술 region 레벨의 LAN에 대응하는 부분의 정보 통신



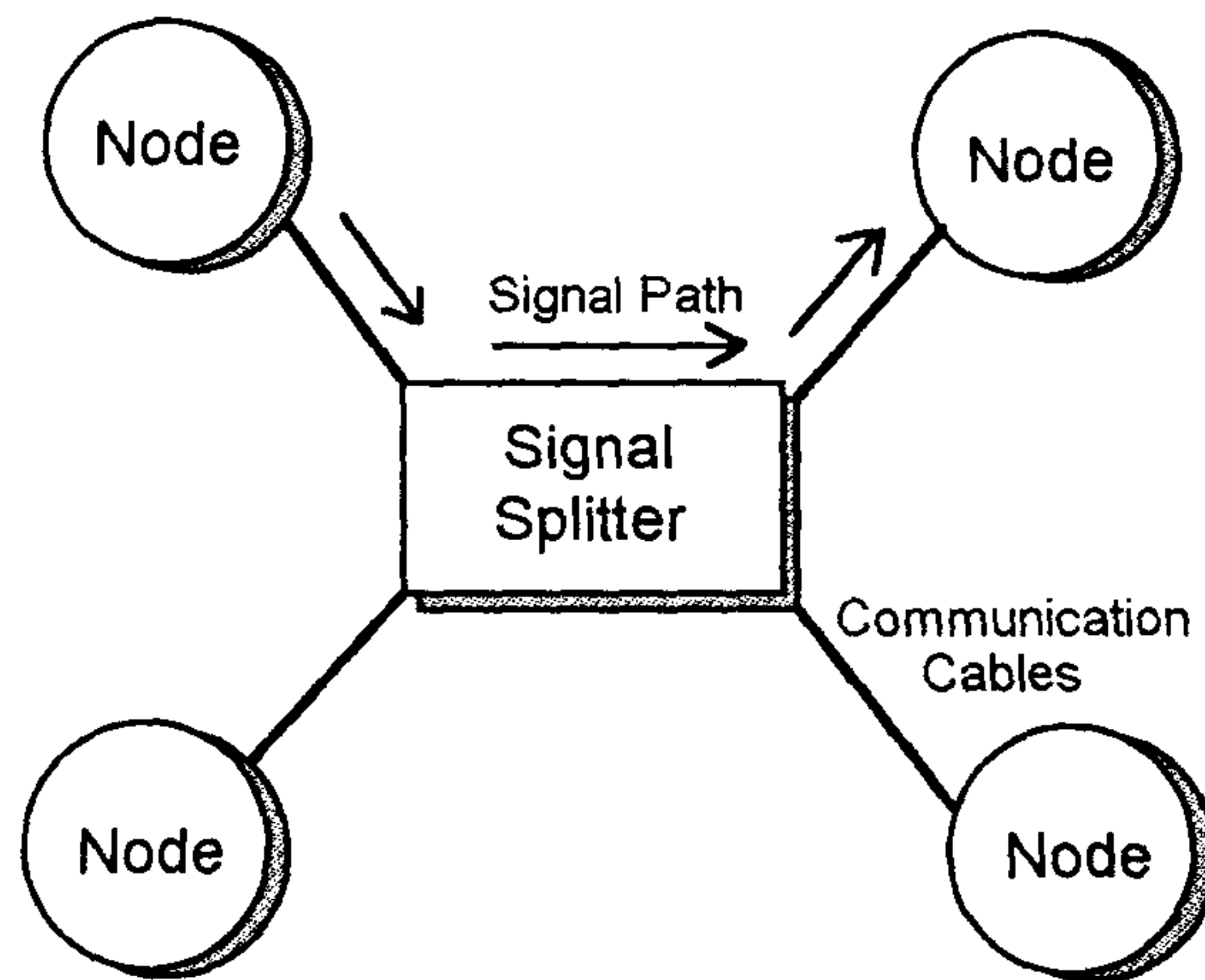
● Star Bus(Arcnet)



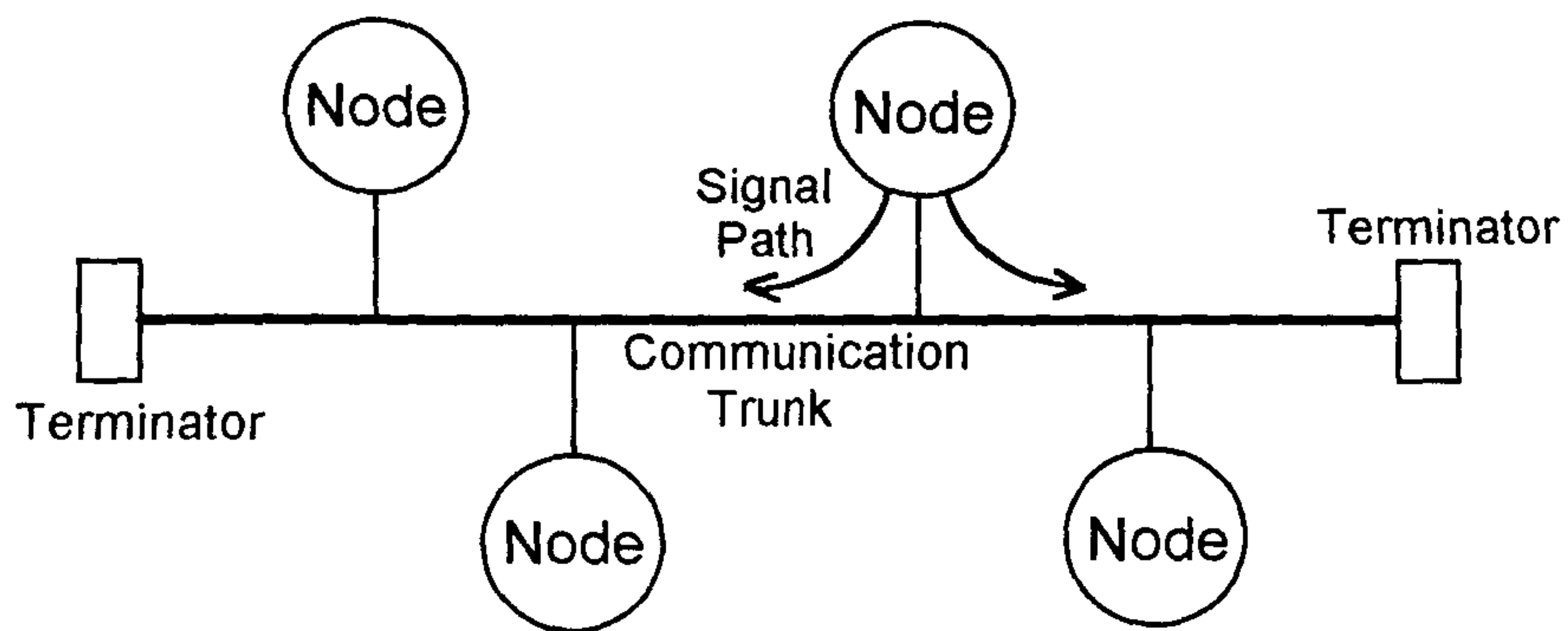
● RING Topology



● STAR RING Topology(IBM Token Ring)



● Linear Bus(EtherNet)



규약을 정의 하고 있다.

그림 2-5는 정보 시스템을 중심으로한 레퍼런스 모델이다. 레벨 1은 각 기계를 제어하는 레벨이고 레벨 2는 스케줄에 따라 기계군에 프로그램 전송 또는 기동하는 것과 동시에 관리에 필요한 정보를 입수하는 프론트 컴퓨터 레벨이다. 레벨 3은 레벨 1과 레벨 2를 포함한 레벨이다. 레벨 4는 여러개의 셀을 포함한 관리 시스템에 대응하는 레벨로써 레벨 3과의 정보 상호 통신을 Technical Region LAN을 통해서 한다. 레벨 5는 여러개의 섹션/에리어를 포함한 레벨이며 레벨 6은 여러개의 Facility Plant를 포함한 레벨이다.

각 레벨은 LAN에 의한 네트워크로 계층화되어 구성된다.

본 연구에서는 CAD/CAM을 중심으로 상위 레벨에서의 Management LAN과 기술 부분 LAN 만을 고려하고 각 기계 등의 하위 레벨에서의 MAP은 고려되지 않았다.

CAD/CAM을 중심으로 한 통합을 위해 본 연구에서는 그림 2-6에서 보는 바와 같이 Sun 670을 파일 서버로 두고 Server & Client 방식으로 CAD 또는 CAD/CAM 시스템을 구성하였으며 전체 네트워크의 기본은 Local Network로써 산업용 LAN(Norm IEEE 802.3) 표준이고 Topology는 Linear Bus 형태의 EtherNET을 이용하였으며 프로토콜은 TCP/IP이다.

또한 PC CAD 시스템으로 부터의 정보와 기타 설계 기준정보에서의 데이터 공유를 위해 PC-NFS(Network Files System)를 통해 호스트의 파일 시스템을 이용하였다.

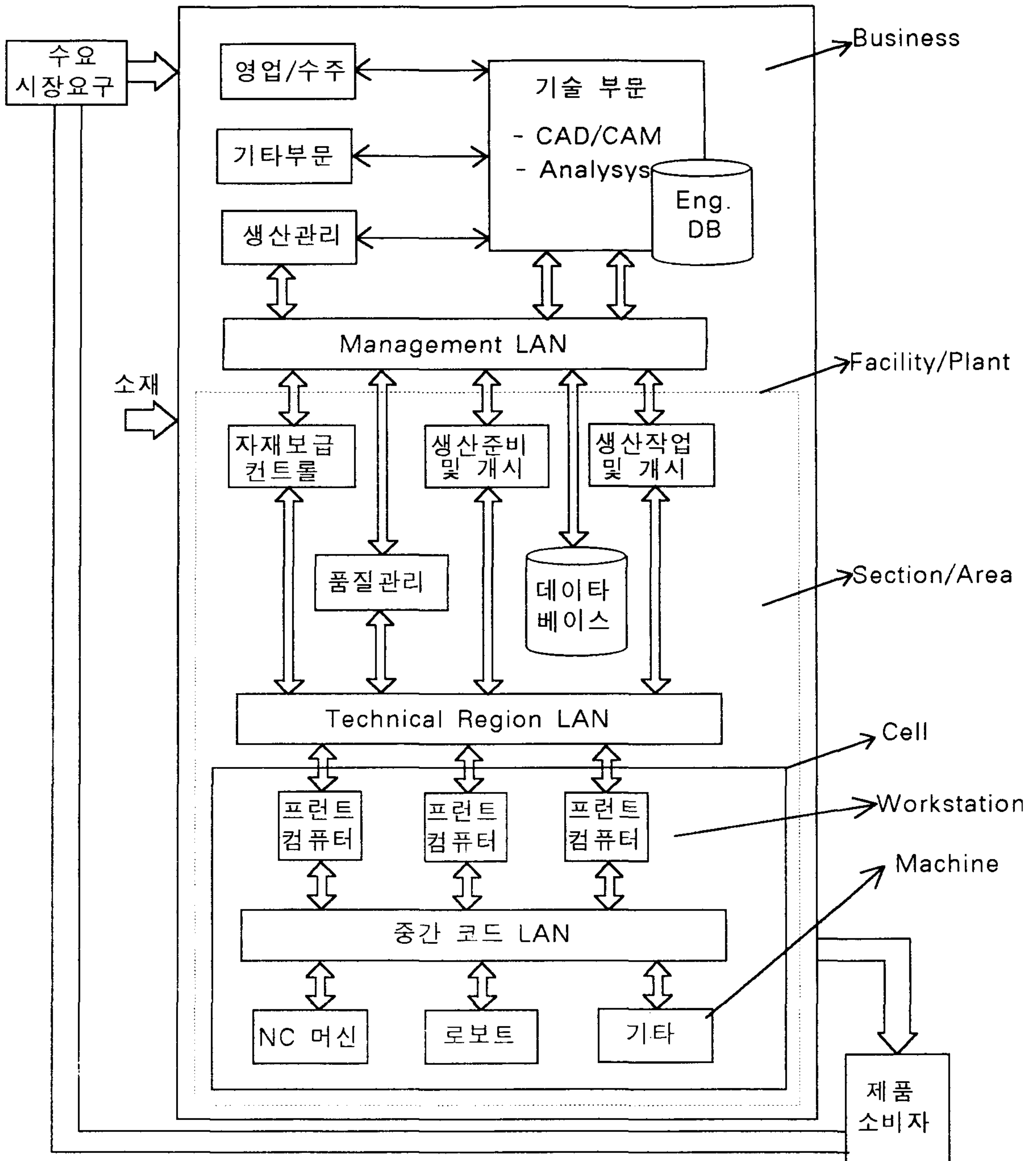


그림 2-5. 시스템 레퍼런스 모델



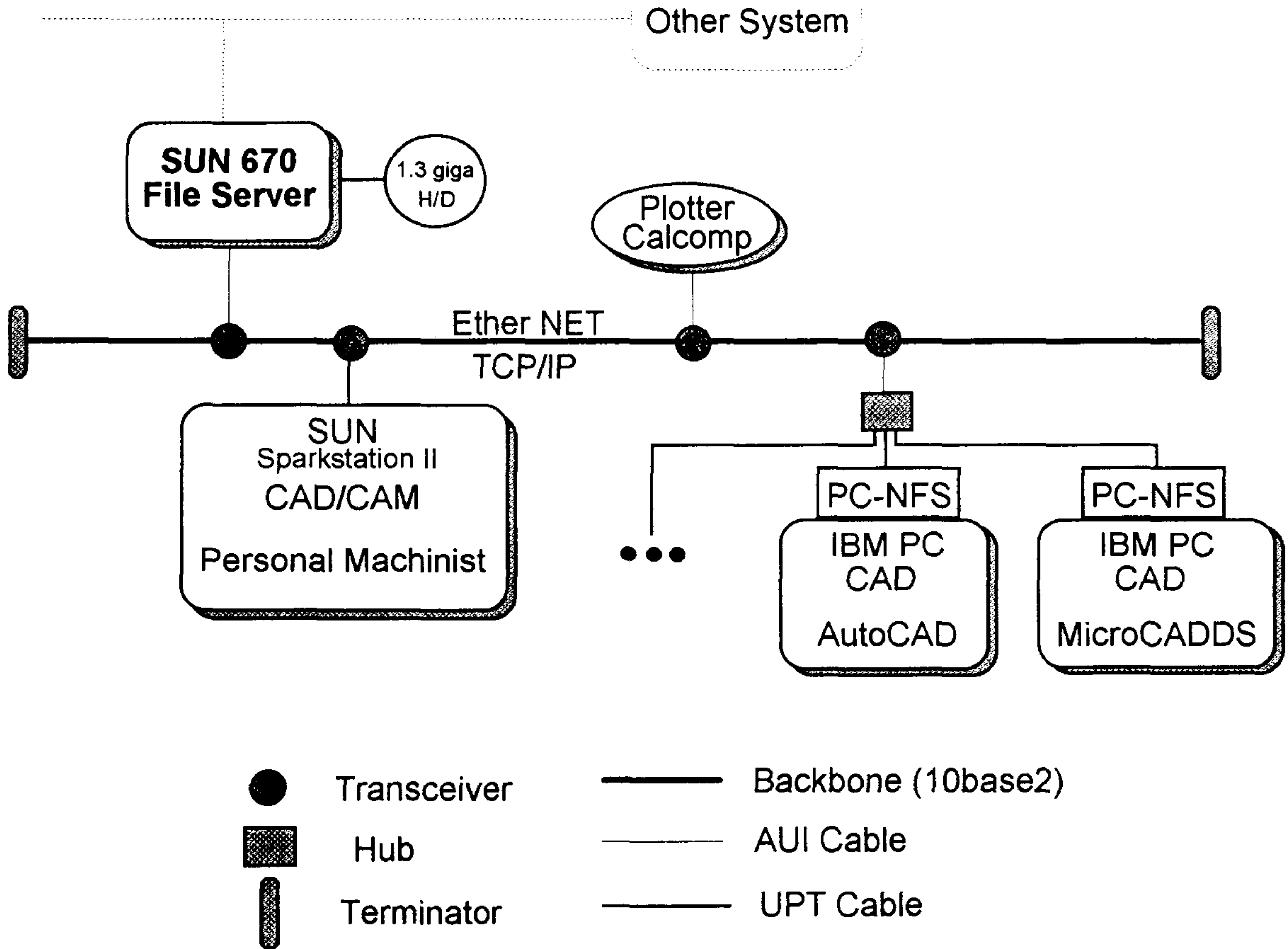


그림 2-6. CAD/CAM 네트워킹

## 2.2. 설계로 부터의 기준 정보 생성

설계와 가공을 일원화하기 위해서 설계에서 많은 정보가 생산을 위해 마련되어야 한다. CAD 내의 도면 작업으로 제품이나 조립품에 대한 부품들이 결정되어지고 이에 따라 BOM(Bill of Materials)을 정의하기 위한 핵심 정보들이 만들어짐으로써 설계로 부터 바로 생산 계획 및 통제 시스템을 위한 기준 정보가 만들어진다.

생산에서 요구되는 생산 공정의 흐름에 따른 정보는 CAD측에서의 설계 정보와는 다른 많은 것을 포함하게 된다. 따라서 본 연구에서는 설계와 생산에서 요구되는 기준 정보를 서로 구분된 구조 관계로 정의하고 이를 동시에 관리할 수 있도록 하였고 부품을 그릴 때 얻어질 수 있는 데이터를 자동으로 얻음으로써 데이터 수집의 수고를 최소화 할 수 있도록 하였다.

물론 CAD로 부터의 정보가 실제 모든 정보들을 다 가질 수는 없기 때문에 이들 기준 정보는 생산 계획 및 통제 시스템에서 더 추가 준비되어야 한다.

설계 단계에서 설계자는 기하학적 형상에 대한 기본 엔터티와 표준 부품의 라이브러리에 액세스하여 부품을 설계하는데 반복을 최대한 피하는 것이 CAD 시스템의 효율을 높이는 것 중의 하나이기 때문에 개발된 금형 설계 전용 CAD 시스템을 이용하여 설계함과 동시에 같은 부품, 또는 비슷한 부품일 경우 기존의 기준 정보를 그대로 이용하는 것도 매우 중요하다.

이에 본 연구에서는 설계측으로 부터의 기준 데이터를 도면, 부품표 및 일반 기술 정보로 나누어 이를 생성, 관리, 운용하고 호스트 컴퓨터

터의 공통 데이터 베이스 내에서 통합 운용될 수 있게 하였다.

설계로 부터 생산에 넘어갈 자료는 크게

- 설계 도면(Design Drawing)
- 부품 정보(Information for Parts or Assemblies)
- 공정 계획(Process Load)
- 부품표(Partlist)
- 기타 기술 정보(Technical Information)

등이 있다.

우선 설계 도면은 PC상에서 전용 CAD 시스템을 이용해 표준 부품 라이브러리를 통해 부품들이 설계되고 이들에 대한 형상 정보는 앞서 말한 DXF 포맷으로 변환되어 공통 데이터 베이스로 저장된다. 제품 설계를 위한 3차원 모델링 역시 포함된다.

한편 CAD시스템으로 부터의 도면 내 우측 하단에 부품 정보표가 그려지게 되며 이 부품 정보표를 ASCII Text 파일로 변환함으로써 도면으로 부터 부품에 대한 정보를 쉽게 얻어 이를 데이터 베이스에 저장할 수 있다.

이를 위해 본 연구에서는 PC상에서 CAD시스템으로 부터 그려진 도면 내 부품 정보표의 Text 부분을 DXF 파일로 변환하고 이 DXF 파일로 부터 부품 정보를 추출하여 이를 데이터 베이스 파일로 바꾸어 공통 데이터 베이스에 저장하였다.

이 때 부품 정보를 포함한 기술정보에 대한 데이터 필드값은 기업마다의 제품에 따라 달라지기 때문에 임의의 표준 형식을 만들고 차후 수정할 수 있도록 하였다.

이상의 내용에 대한 기본 구조는 그림 2-7과 같다.

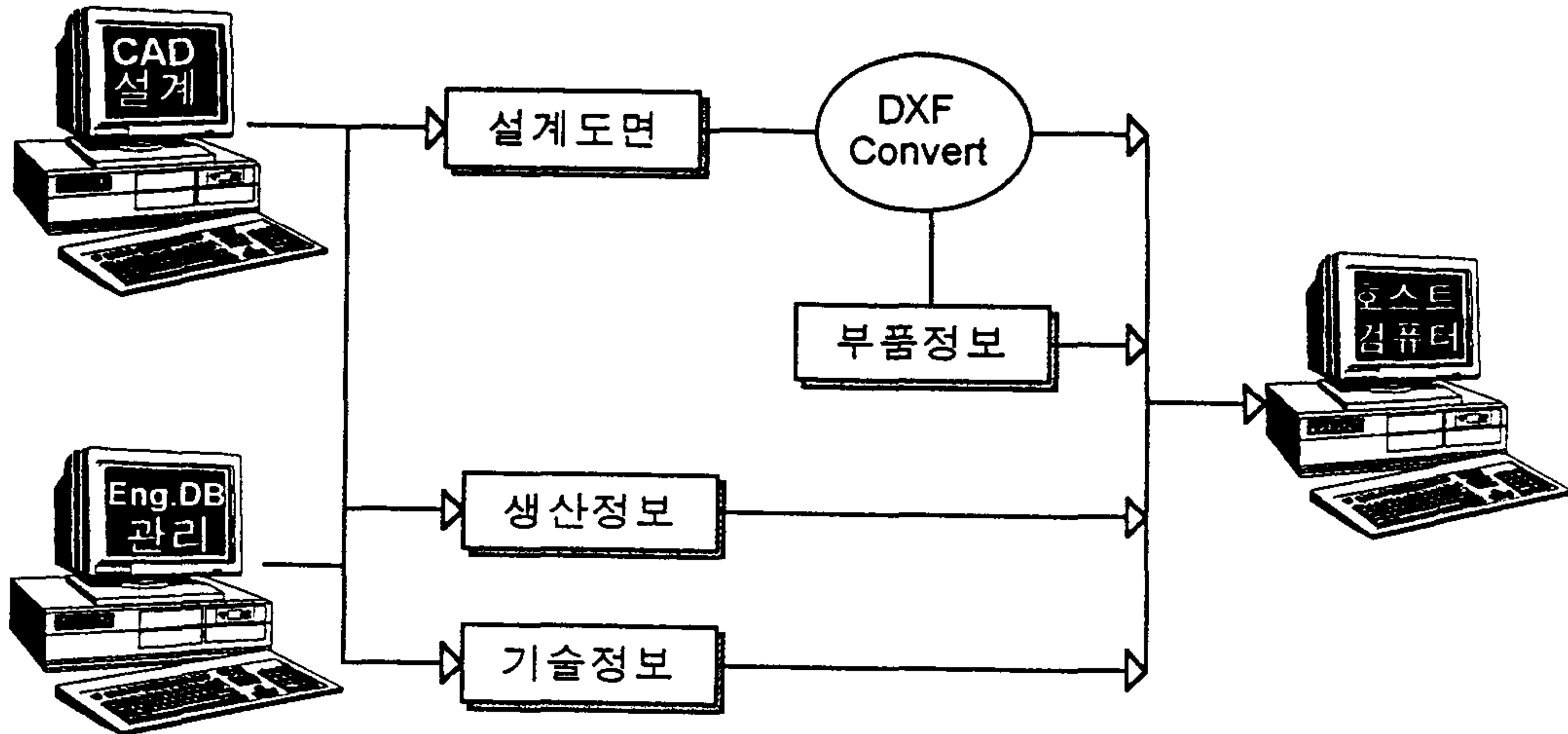


그림 2-7. 설계 기준 정보 생성의 기본 구조

설계 시 각 부품 별로 부품 정보에 대한 데이터 파일이 생성되어 dBASE V의 데이터 베이스 파일로 만들어지고 이는 기준 정보 관리 모듈로 부터 부품 정보나 부품표 및 기타 기술 정보를 설계자가 조회, 수정할 수 있도록 하였다.

이는 일반 범용 PC CAD 시스템과 유연하게 연결하기 위해 IBM PC상에서 Clipper와 C Language를 이용하였으며 LAN상에서 PC-NFS(Network File System)를 이용하여 공통 데이터 베이스와 연결된다. 전체는 5개의 메인 메뉴로 구성되며 다음과 같다.

제품 선택 메뉴에서는 크게 프라스틱 사출 금형과 프레스 금형으로 나뉘어 각 제품에 따라 호스트 컴퓨터의 디렉토리 별로 관리가 되며 이는 NFS에 의해 액세스 된다.

수정 메뉴에서는 기존의 공통 데이터 베이스 내에 있는 제품 정보,

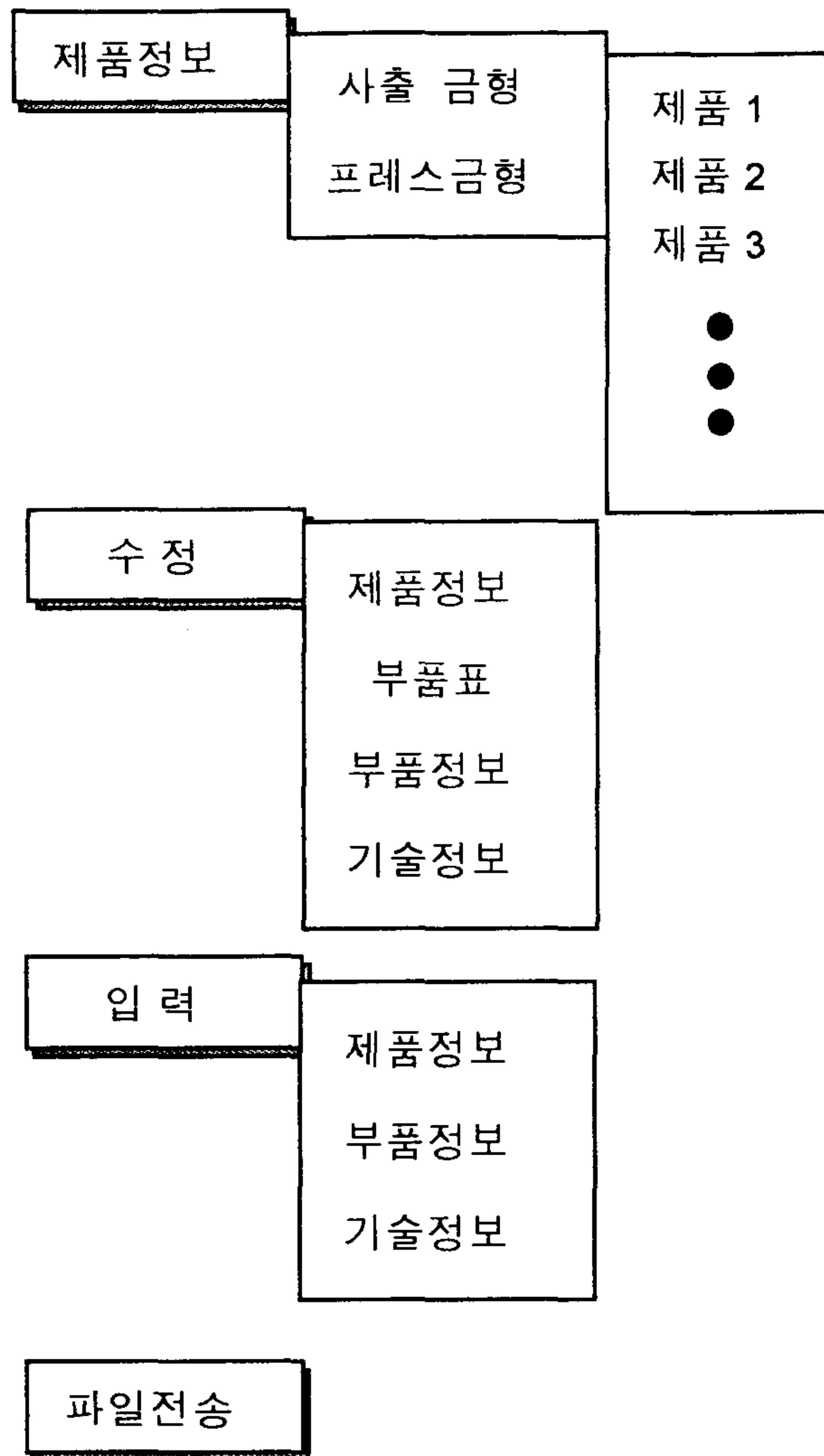


그림 2-8. 기준 정보 생성 모듈 메뉴 구성

부품표, 부품 정보 및 기술 정보에 대한 파일을 불러 설계자가 정보를 조회 및 수정할 수 있다.

입력 메뉴에서는 새로운 제품 및 부품 설계시 제공될 필요가 있는 제품 정보, 부품표 및 기타 기술 정보등을 입력한다. 입력된 정보에 의해 생성된 데이터 파일은 파일 전송 메뉴에서 공통 데이터 베이스로 저장된다.



다음은 본 시스템의 기준 정보 생성 모듈의 한 예를 보여준다.

제품 선택	수정	입력	파일전송	종료
선택 제품 : PHONE * 레코드 삭제 : <Del> Key * 레코드 추가 : <Ins> Key * 레코드 입력 : 해당 위치에 커서를 두고 <Enter> Key				
부품번호	부 품 명	재 질	수 량	비 고
1	upper fitting plate	s55c	1.00	<Memo>
2	upper cavity plate	s55c	1.00	<Memo>
3	lower cavity plate	s55c	1.00	<Memo>
4	support plate	s55c	1.00	<Memo>
5	space block	s55c	2.00	<Memo>
6	ejector retain plate	s55c	1.00	<Memo>
7	ejector plate	s45c	1.00	<Memo>
8	lower fitting plate	s55c	1.00	<Memo>
9	stripper plate	s55c	1.00	<Memo>
10	upper cavity	f-s55c	1.00	<Memo>
11	lower cavity	f-s55c	1.00	<Memo>
19	locate ring A type	sm45c	1.00	<Memo>
20	sprue bush A type	sts3	1.00	<Memo>
새 데이터를 입력하시요 : sm45c				

기존 데이터베이스의 내용을 수정하십시오

그림 2-9. 기준 정보 생성 모듈 화면 예

### 2.3. 기준 정보 운용 관리

PC로 부터의 도면과 기준 정보로 부터 호스트 컴퓨터에서는 이를 통합 관리 운용한다.

본 시스템에서는 설계로 부터의 기준 정보와 NC 가공 데이터를 포함한 가공 정보가 공통 데이터 베이스에 저장하고 각 제품별로 트리 형태의 디렉토리를 구성하여 도면 데이터 파일, 가공 데이터 파일 및 기술 정보 파일을 갖고 운영된다.



이들 각 파일들은 네트워크에 의해 연결된 모든 설계자들에 의해서 앞서 기술한 PC측의 기준 정보 생성 모듈로 부터 만들어지며 크제는 설계 도면 관리를 위한 모듈과 설계/가공 기술 정보 관리 모듈 및 데이터 파일 수정, 삭제 등의 유틸리티 모듈들로 구성된다. 다음 그림 2-10은 그 구성도이다.

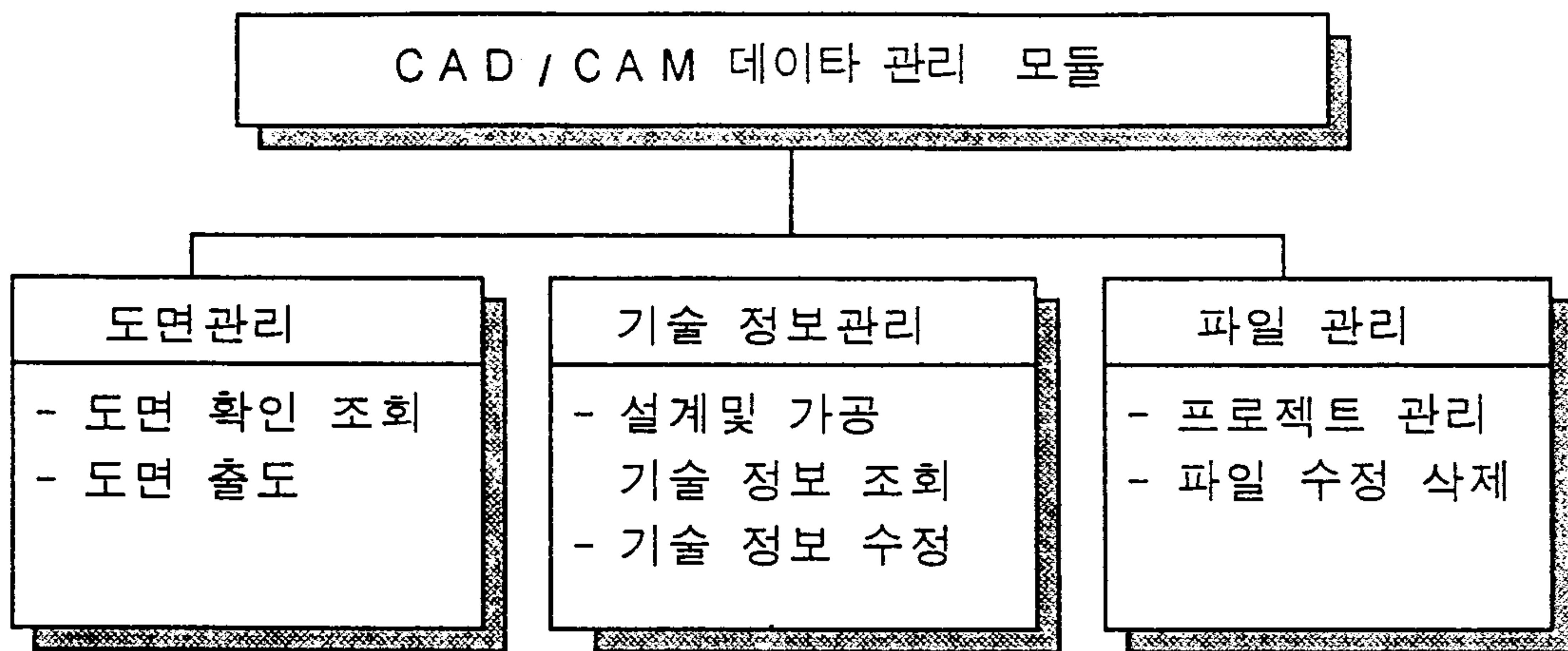


그림 2-10. 기준 정보 운용 관리 모듈 구성도

도면 관리 부분은 제품별로 정리된 부품의 리스트로 부터 사용자가 쉽게 선택해 도면을 확인하고 수정 작업을 할 수가 있으며 필요시 도면 출도가 가능하다.

기준 정보 관리 부분에서는 설계로 부터의 기준 정보를 조회, 수정 작업을 할 수 있으며 이는 차후 생산 일정 계획등 생산 계획 및 통제 시스템에서의 추가 데이터와 함께 통합된 기준 정보 자료로써 이용될 수 있다.

설계로 부터의 기준 정보는 기업마다의 제품에 따라 각기 달라질 수

있기 때문에 본 연구에서는 기본적인 요소만 고려되고 경우에 따라 확장할 수 있게 되었다.

파일 관리 부분에서는 공통 데이터 베이스 내에 도면 파일이나 기술 정보 파일 등의 데이터 파일들을 조회, 수정, 삭제할 수 있고 임의의 프로젝트 종결시 필요에 따라 전체의 내용을 모두 지울 수 있도록 하였다. 물론 보안이 관리되며 확인작업을 거친다.

본 시스템은 UNIX의 파일 구조를 갖고 제품별 트리 형태의 디렉토리로 관리하기 때문에 이들 구조상에서 각 디렉토리에 따라 임의의 permission을 부여함으로써 데이터에 대한 보안을 관리할 수 있도록 하였다.

이들은 C Language를 이용하였고 생산 현장의 이용자의 편의를 충분히 고려하여 유저 인터페이스는 XVIEW(X Window-System-based Visual/Integrated Environment for Workstation) 라이브러리를 통한 GUI(Graphic User Interface)를 이용하여 구축되었다.

XVIEW는 X Window 시스템 하에서 운용되는 interactive한 그래픽 베이스의 응용 어플리케이션을 지원하기 위한 유저 인터페이스 툴킷으로 모든 자원이 한 패키지로 부터의 오브젝트 개념을 이용한 Object-Oriented 프로그램이다.

다음 그림 2-11은 XView의 클래스의 계층 구조를 보여 준다.

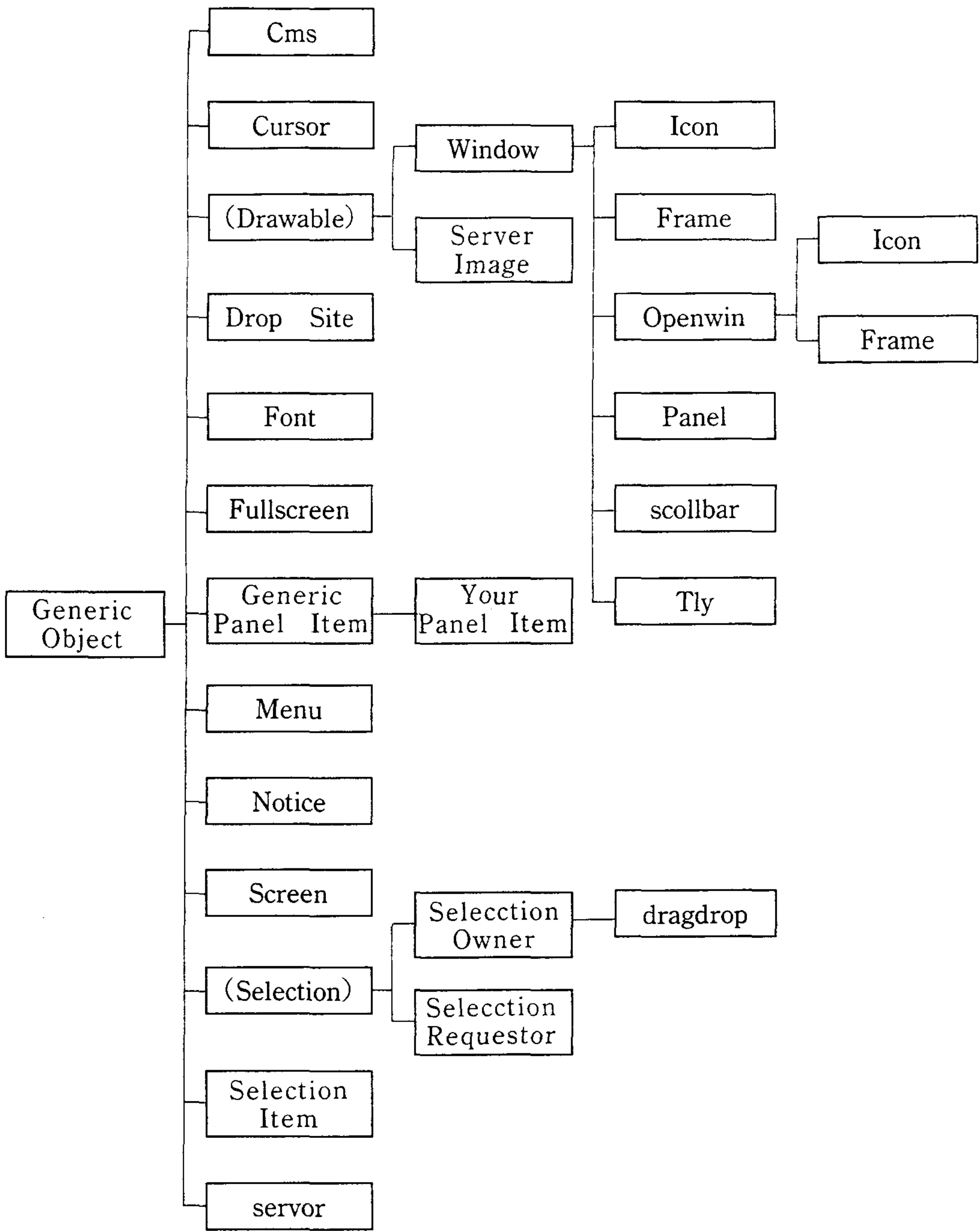


그림 2-11. Xview 클래스 계층 구조

다음 그림 2-12과 2-13은 본 시스템의 기준 정보 관리 화면 예를 보여준다.



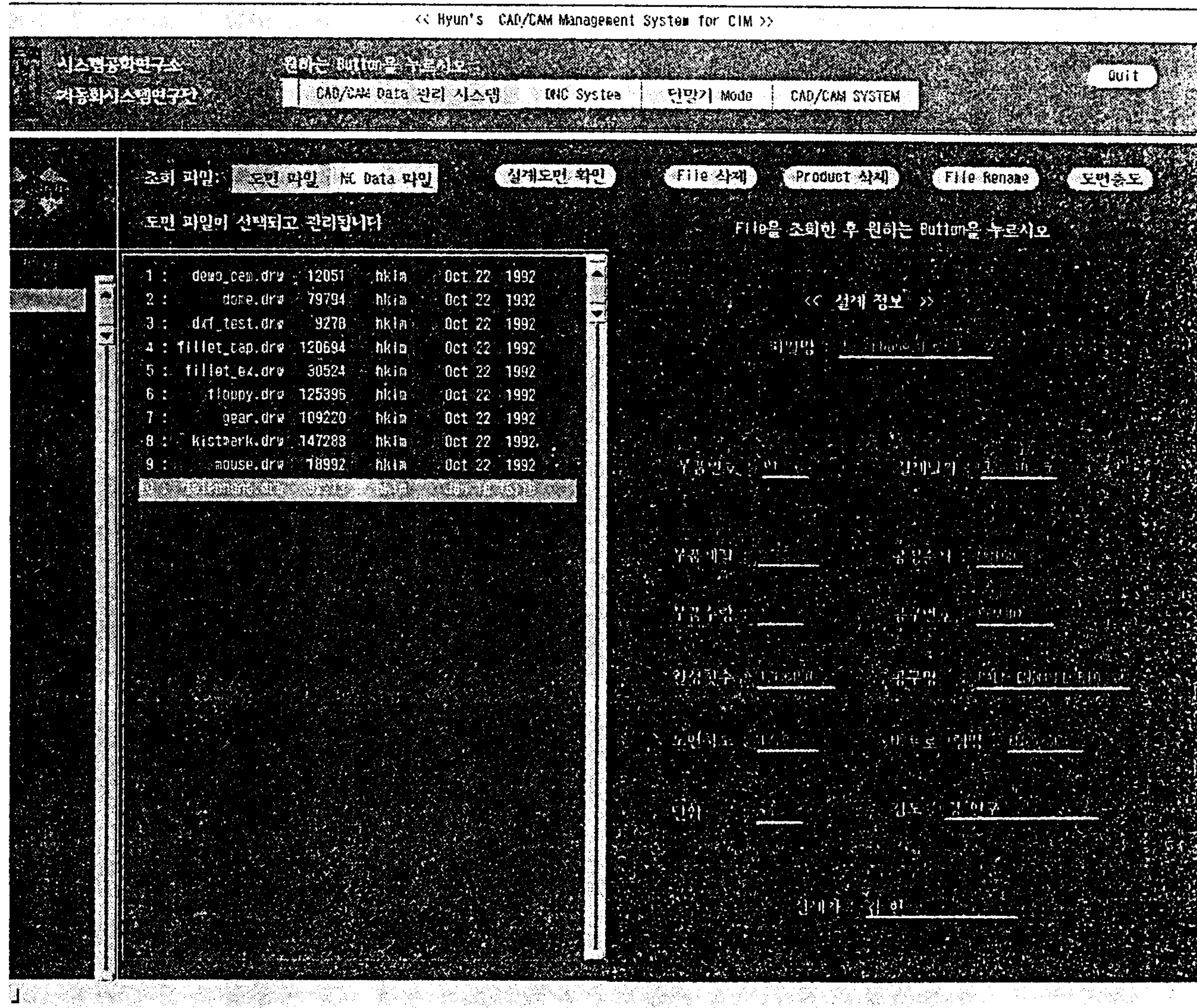


그림 2-12. 기준 정보 관리 모듈의 화면 예  
(기준 정보 조회)



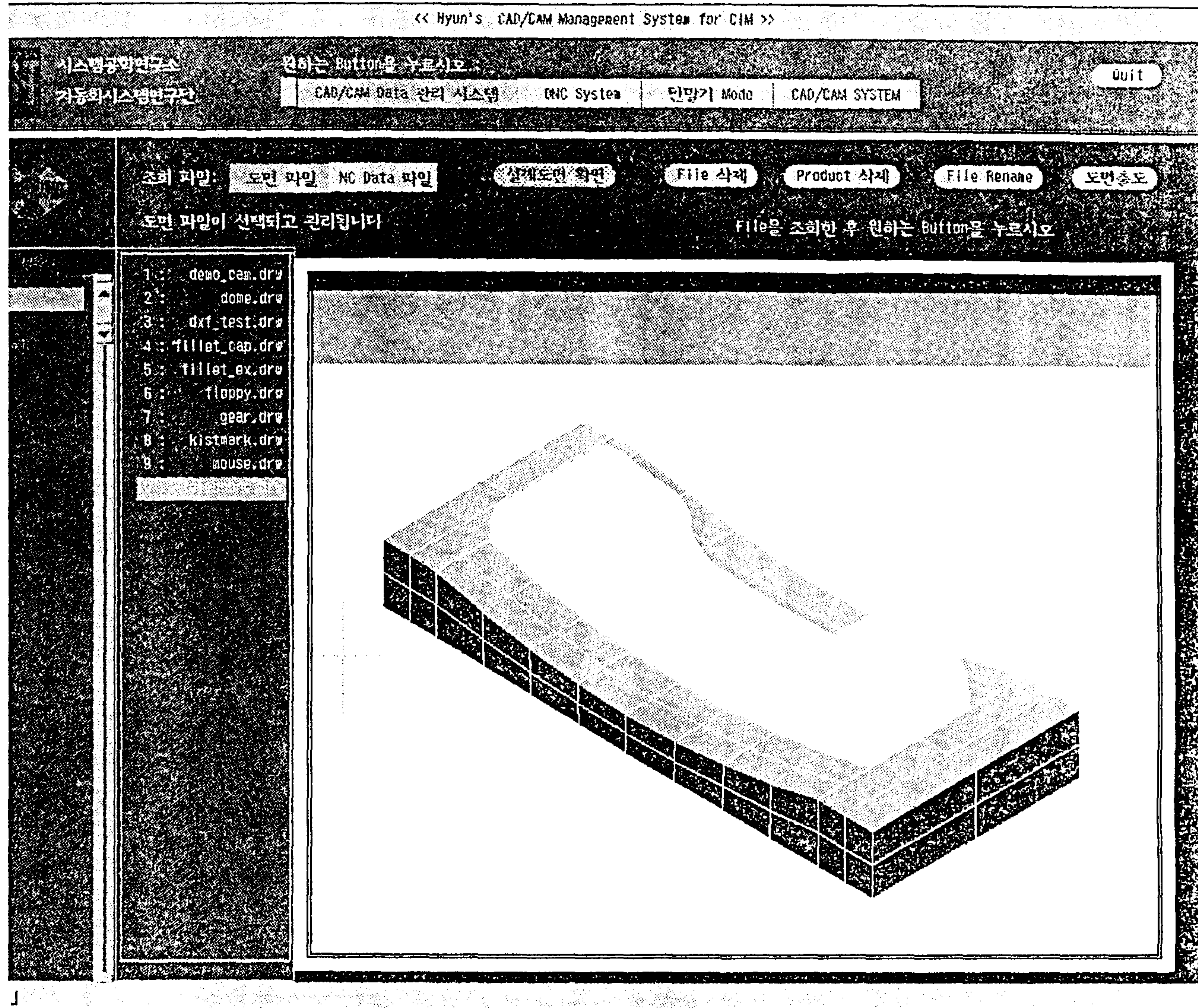


그림 2-13. 기준 정보 관리 모듈의 화면 예  
(도면 관리)



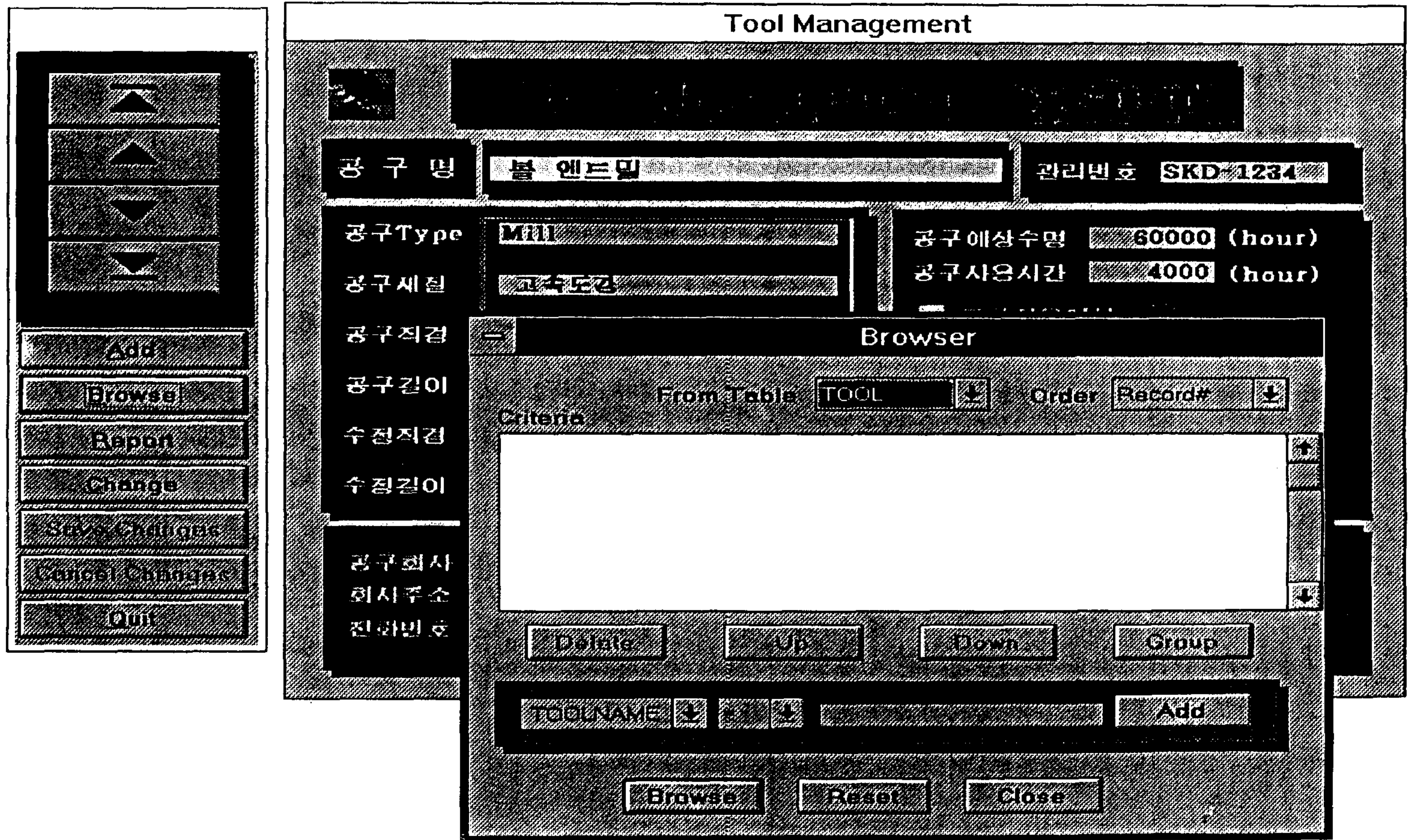


그림 2-14. 기준 정보 관리 모듈의 화면 예  
(공구 관리)



통합 데이터 베이스에 기하학적 형상 정보는 현재로서는 아직 표준 데이터 베이스에 부응되지 않기 때문에 CAD 소프트웨어를 이용 중간 파일 형태로 관리되었다. 물론 기하학적 형상 데이터를 표준 데이터 베이스로 해결 할 수 있는 새로운 D/B 시스템을 개발하거나 S/W 모듈을 이용하여 다른 응용 프로그램에 확장시킬 수 있는 코어를 정의함으로써 이 문제를 해결할 수도 있을 것이다.

종래의 관리 방법은 한 Workplace에서 CAD/CAM과 생산 계획 및 통제 시스템이 따로 갖춰져 서로의 데이터의 전송이 매뉴얼로 행해졌으나 본 시스템은 파일 서버에 공통 데이터 베이스가 관리 운용될 수 있도록 구성함으로써 기타 다른 모듈이 추가 연계될 때 특히 생산 계획 및 통제 모듈과 하나로 연계된 시스템으로 확장할 수 있다.

따라서 BOM, CAD 데이터, NC 데이터 및 생산 일정등의 데이터들이 서로 독립되었지 않고 CAD/CAM과 생산 계획 및 통제 시스템에서 같은 데이터 베이스로 액세스하여 사용하며 어느 한쪽에서의 데이터 갱신이 즉각 다른 쪽에서 이용될 수 있기 때문에 고도의 통합 개념에 크게 접근하고 있다.

이러한 통합 레벨에서는 CAD/CAM이나 PPC 양쪽을 위한 통합된 데이터 구조가 정의되어야 할 뿐만 아니라 통합된 데이터 베이스 시스템이 설치 되어야 할 것이다. 따라서 현재 연구단에서 추진 중인 계획 및 통제 시스템 개발 역시 이러한 방향으로 통합될 것이다.

### 제3절 DNC 시스템 개발

#### 3.1. CAD와 CAM의 연계

종래 도면으로 부터 APT등의 프로그램을 이용하거나 또는 copy milling등에 의한 가공 데이터 산출 방법이 설계에서의 CAD 활용이 커지고 대중화됨에 따라서 CAD에서의 모델링 정보를 이용하여 가공 데이터를 산출하는 CAD/CAM 시스템들이 크게 각광을 받고 있다.

CAD/CAM이란 제품의 설계와 제작에 컴퓨터를 원용하는 것을 의미하지만 CAD/CAM 시스템은 컴퓨터 그래픽스 기능을 이용한 하나의 소프트웨어 시스템을 의미하는데 현재 국내에서도 대형 CAD/CAM 시스템이 200여개 이상 도입되어 활용되고 있다.

이러한 CAD/CAM 시스템에서 CAD와 CAM간의 데이터 연계는 기하학적인 형상 정보와 CAD 내의 기술 정보로 부터의 가공 정보 및 생산 공정 계획을 얻어 설계와 가공을 일원화 하려는 것이다.

실제 기하학적 형상 데이터는 CAD 시스템의 데이터 구조로 부터 얻어지며 이들의 모든 기하학적 정보가 가공에 다 관련된 것이 아니기 때문에 다시 가공에 필요한 데이터를 얻어 변환할 필요가 있다.

또한 CAD 시스템과 CAM 시스템이 별개로 존재할 때는 CAD 시스템으로 부터 IGES 등의 중간 파일 형식으로 데이터를 전송받아 기하학적 정보를 얻는다.

모든 가공이 원칙적으로 3차원 정보를 필요로 하기 때문에 즉 기하학적 형상의 완벽한 변환을 위해서는 3차원 CAD 시스템을 필요로 하

기 때문에 일반 설계에서의 2차원 CAD 시스템과 유한요소 해석이나 가공 데이터 산출을 위한 3차원 CAD 시스템이 혼용되어 이용되는 것이 가장 경제적이다.

CAD/CAM 시스템으로 부터 얻어진 가공 데이터 또는 APT 등의 가공 프로그램은 각 기계의 특성, 공구, 소재등에 맞도록 포스트 프로세스 과정을 거쳐 가공 스케줄 및 기타 가공에 필요한 데이터와 함께 NC 가공 기계에 전달된다. 그림 2-15는 이러한 과정을 간략하게 보여준다.

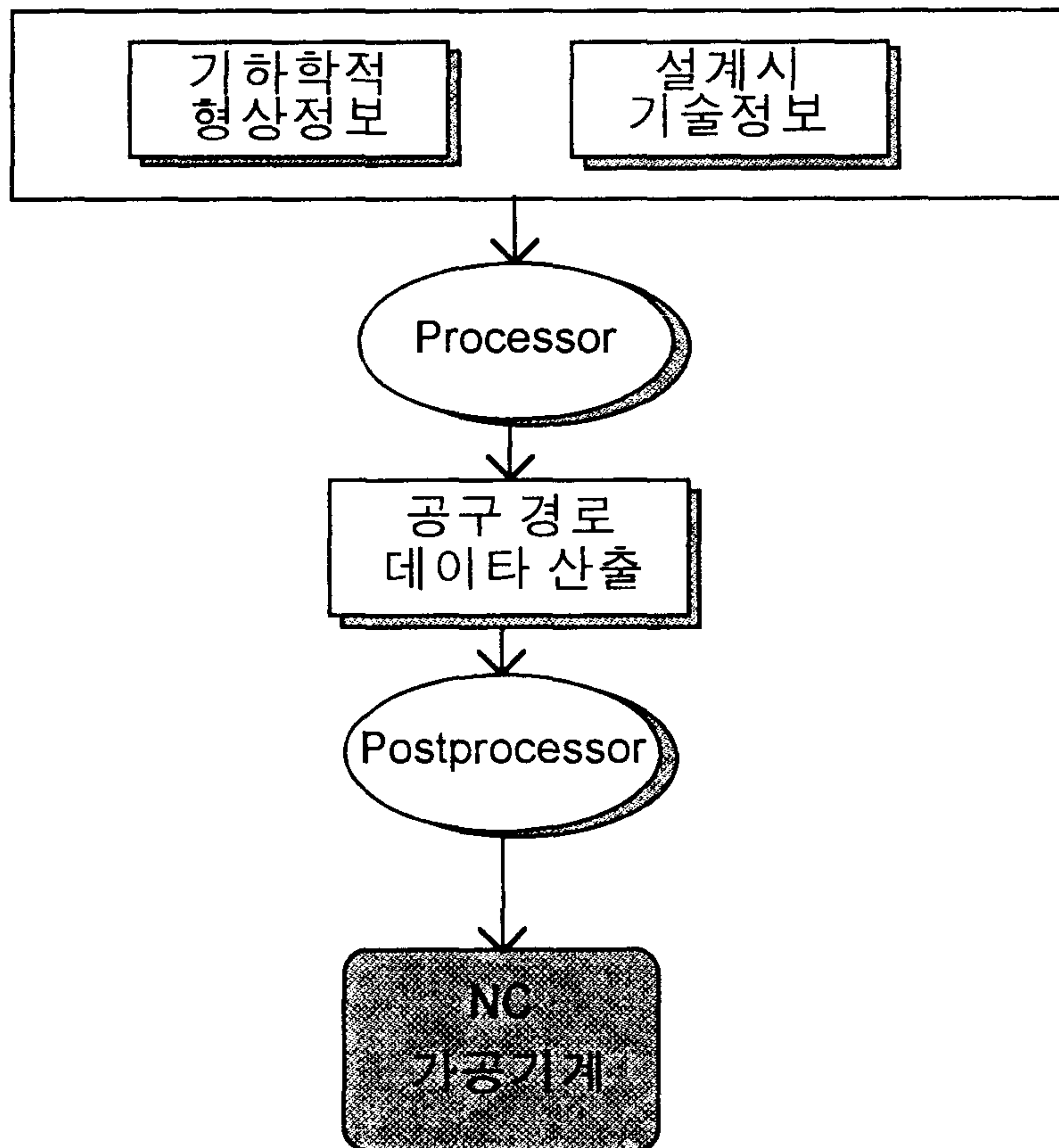


그림 2-15 설계/가공의 일원화 프로세스

### 3.2. CL 데이터 산출 및 포스트 프로세서 개발

본 시스템에서는 간단한 2차원 형상에 대한 방전 가공을 위해 2차원 CAD 설계로부터의 기하학적 형상을 DXF 파일 포맷으로 얻고 이에 대한 공구 경로 결정 및 원시 데이터를 생성하며, 이를 미쓰비시 방전 가공기에 맞도록 포스트 프로세스함으로써 NC 가공 기계와 직접 연결할 수 있도록 하였다.

한편 3차원 자유 곡면 등의 윤곽 가공은 범용 CAM 소프트웨어를 이용하여 CL(Cutter Location) 데이터를 얻고 이를 기계에 맞게 후처리 할 수 있는 포스트 프로세서를 개발하여 NC 가공 기계와 연결하였다.

그림 2-16은 CL 데이터와 CC(Cutter Contact) 데이터를 보여준다.

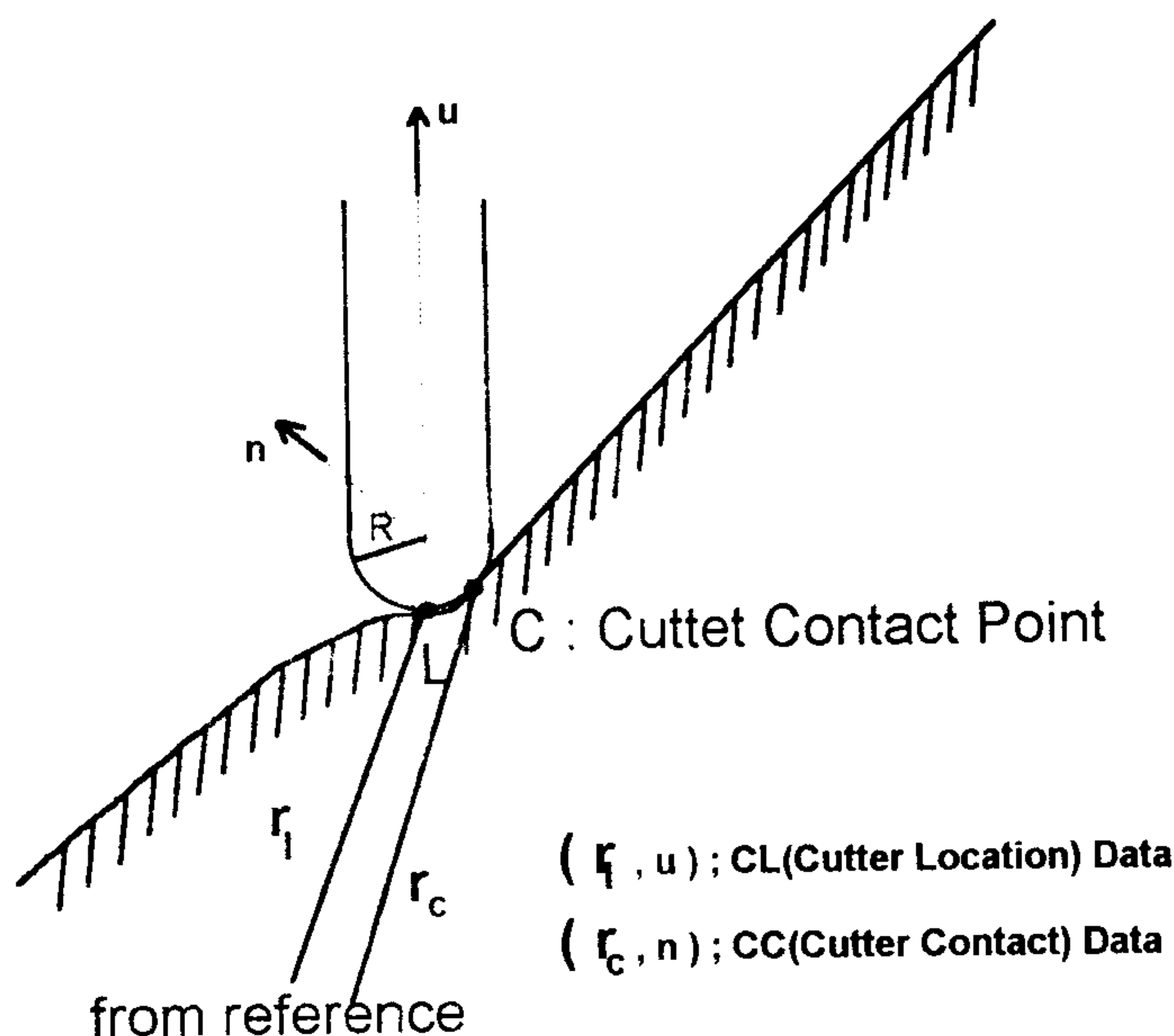


그림 2-16 CL(Cutter Location) 데이터



일반적인 3차원 CAD 시스템으로 부터의 기하학적 형상 정보는 DXF 또는 IGES 파일 포맷으로 데이터를 얻어 이를 범용 CAM 시스템을 위한 입력으로 활용하며 2차원 CAD 시스템에서의 와이어 프레임 모델은 3차원 CAD 시스템에서 면처리 작업을 한 후 같은 방법으로 CAM 시스템과 연결된다. 데이터간의 통신은 앞서 언급된 네트워크에 의해 공통 데이터 베이스에 액세스하여 쉽게 행해질 수 있다.

포스트 프로세서는 CAM 시스템으로 부터 나온 CL 데이터 또는 원시 데이터를 각 기계 특성에 맞도록 변환하는 작업을 하게 되는데 이를 위해서는 각 기계 특성에 따른 많은 정보가 요구된다.

이들 정보는 CL 데이터와 함께 전형적인 NC 코드인 어드레스와 4자리의 수치값으로 변환됨으로써 후처리된다. 이들의 대표적인 어드레스는 표 2-1와 같다.

표 2-1. NC 코드 어드레스

기능	어드레스	의미
Program Number	: , O	Program number
Sequence Number	N	Sequence number
Preparatory Function	G	Motion mode(Linear,Arc...)
Dimension Word	X,Z,U,W R I,K	Coordinate axis motion Arc radius, Corner R Coordinate of arc center
Feed Function	F,E	Feedrate, Thread lead
Spindle Speed Function	S	Spindle speed
Tool Function	T	Tool no., Tool offset no.
Miscellaneous Function	M	On/Off control on M/C
Dwell	P,U,X	Dwell time
Program # designation	P	Designation of subprogram
Sequence # designation	P,Q	Designation of sequence no. of Repeated program
Repetitive Count	L	Subprogram repetitive count
Parameter	A,D,I,K	Parameters in canned cycle0

본 연구에서는 포스트 프로세서의 개발을 위해 기본적인 NC 코드는 EIA Standard(Electric Industries Association) RS-273에 준하였고 표 2-2는 EIA Standard RS-273에서의 G-Code를 보여 준다.

이들 EIA-Standard G-Code 기능은 일반 NC 공작 기계에서의 제공되는 기능과 다소간의 차이를 보일 수 있다. 따라서 본 연구에서는 표 2-3과 같은 사항들이 고려되어 포스트 프로세서를 개발하고 몇가지 기계에 대해 검증하였다.

표 2-2. EIA-standard에서의 G Code

Code	Function
G00	Positioning
G01	Linear Interpolation
G02	Circular Interpolation(CW)
G03	Circular Interpolation(CCW)
G04	Dwell
G05	Unsigned
G06	Parabolic Interpolation
G07	Unsigned
G08	Acceleration code(exponential rate)
G09	Deceleration code(exponential rate)
G10   G12	Unassigned
G13   G16	<operate particular set of axes>



Code	F u n c t i o n
G17	XY-plane specification
G18	ZX-plane specification
G19	YZ-plane specification
G20   G32	Unassigned
G33	Constant lead thread cutting
G34	Increasing lead thread cutting
G35	Decreasing lead thread cutting
G36   G39	Unassigned
G40	Cutter compensation cancel
G41	Cutter compensation left
G42	Cutter compensation right
G43	<cutter offset (inside corner)>
G44	<cutter offset (outside corner)>
G45   G49	Unassigned
G50   G59	Reserved for adaptive control
G60   G69	Unassigned
G70	inch Programming
G71	mm Programming
G72	3-D Circular Interpolation (CW)
G73	3-D Circular Interpolation (CCW)

Code	F u n c t i o n
G74	Multiquadrant circular interpolation cancel
G75	Multiquadrant circular interpolation
G76   G79	Unassigned
G80	Canned cycle cancel
G81	Drilling cycle
G82	Drilling cycle with Dwell
G83	Peck drilling cycle
G84	Tapping cycle
G85	Boring cycle
G86	Boring cycle
G87	Back boring cycle
G88	Boring cycle
G89	Boring cycle
G90	Absolute input
G91	Incremental input
G92	Work coordinate system change
G93	Inverse time feedrate
G94	feed per minute
G95	feed per revolution
G96	Constant surface speed per minute
G97	Spindle speed in revolution per minute
G98	Unassigned
G99	Unassigned

표 2-3. 포스트 프로세서 개발시 고려 사항

고 려 사 항	비 고
- 좌표계	Absolute, Incremental
- 단위계	Inch, Metric
- 소숫점 자리수 표현	N5, N005
- 프로그램 번호 지정 유무	Yes/No
- Block Number 자릿수 표현	+012.400, 12.400, 12.4
- 마지막 Block(End of Block) 커맨드	\$, *, NULL
- Block당 쓰여질 수 있는 G code 갯수	G**   G**   G**   ....
- Initial Sequence	G90   G40   M05   M09
- Shutdown Sequence	G28   M05   M09
- Tape Start Code	%
- Tape End Code	%
- End of Tape	M02, M30
- Setup position	G92, G50, G98
- 급속 이송을 위한 G code	G00
- 급속 이송 시 feedrate 지정 유무	F100, F250, NULL
- 급송 이송, 절삭이송시 단위 변경 유무	IPM, MMPM
- 드웰 표시 문자	D or F
- 원호 지원	90° 이하, 360°
- 원의 표현 방식	R or I,J,K
- 최대 반경	real value
- 공구 원호 보정 유무	G40/G41/G42
- 공구 길이 보정 유무	G43/G49
- 공구 번호 자릿수	4(2,2)
- 표준 4자리 공구번호 사용여부	index pos/offset no.
- ATC 유무	Yes/No
- 공구 교환 코드	M06
- Tool Stop Sequence	M09   M05   M01
- 공구 로딩후 첫 명령	M03   M08
- 공구 로딩 후 새로운 preset	G92, G50
- 스피들 속도 자릿수	S****
- 스피들 속도 표현	S1000
- 최대 스피들 속도	8000

본 연구에서는 간단한 Modelling Machine인 Roland사의 CAMM-3 Model PNC-3000을 밀링용으로 이용하였으며 이는 일반 G-Code를 쓰지 않고 자체 포맷을 갖고 있어 이에 따른 포스트 프로세서를 만들어 실제 가공하였고 FAPT 프로그램을 사용하는 FUNUC System P-Model G로 NC 데이터 산출의 결과를 마찬가지로 검증하였다.

다음은 본 연구에 활용된 CAM 소프트웨어인 Computer Vision사

G0 X0.00000 Y0.00000 Z0.20000	H
G1 X0.00000 Y0.00000 Z0.05063 F10	V 22
X7.50000 Y0.00000 Z0.05063	Z0.0, 0.0, 0.0
X7.50000 Y0.43750 Z0.05063	Z0.0, 0.0, -24.68
X0.00000 Y0.43750 Z0.05063	V10.0
X0.00000 Y0.87500 Z0.05063	Z7500.0, 0.0, -24.68
X7.50000 Y0.87500 Z0.05063	Z7500.0, 437.5, -24.68
X7.50000 Y1.31250 Z0.05063	Z0.0, 437.5, -24.68
X0.00000 Y1.31250 Z0.05063	Z0.0, 875.0, -24.68
X0.00000 Y1.75000 Z0.05063	Z7500.0, 875.0, -24.68
X7.50000 Y1.75000 Z0.05063	Z7500.0, 1312.5, -24.68
X7.50000 Y2.18750 Z0.05063	Z0.0, 1312.5, -24.68
X0.00000 Y2.18750 Z0.05063	Z0.0, 1750.0, -24.68
X0.00000 Y2.62500 Z0.05063	Z7500.0, 1750.0, -24.68
X7.50000 Y2.62500 Z0.05063	Z7500.0, 2187.5, -24.68
X7.50000 Y3.06250 Z0.05063	Z0.0, 2187.5, -24.68
X0.00000 Y3.06250 Z0.05063	Z0.0, 2625.0, -24.68
X0.00000 Y3.50000 Z0.05063	Z7500.0, 2625.0, -24.68
X7.50000 Y3.50000 Z0.05063	Z7500.0, 3062.5, -24.68
X7.50000 Y3.93750 Z0.05063	Z0.0, 3062.5, -24.68
X0.00000 Y3.93750 Z0.05063	Z0.0, 3500.0, -24.68
X0.00000 Y4.37500 Z0.05063	Z7500.0, 3500.0, -24.68
X7.50000 Y4.37500 Z0.05063	Z7500.0, 3937.5, -24.68
X7.50000 Y4.81250 Z0.05063	Z0.0, 3937.5, -24.68
X0.00000 Y4.81250 Z0.05063	Z0.0, 4375.0, -24.68
X0.00000 Y5.25000 Z0.05063	Z7500.0, 4375.0, -24.68
X7.50000 Y5.25000 Z0.05063	Z7500.0, 4812.5, -24.68
X7.50000 Y5.68750 Z0.05063	Z0.0, 4812.5, -24.68
	Z0.0, 5250.0, -24.68
	Z7500.0, 5250.0, -24.68
	Z7500.0, 5687.5, -24.68

(a) CL Data 예

(b) CAMM-3 입력데이터

그림 2-17 CL 데이터와 CAMM-3 입력 데이터 비교



의 Personal Machinist에 의한 CL Data와 이를 CAMM-3 Model PNC-3000에 맞게 포스트 프로세싱한 보기를 보여준다.

### 3.3. DNC System 구성

설계에서 CAD/CAM 도입이 활성화되고 마이크로 프로세서 및 컴퓨터가 가공 공정의 많은 부분을 지원하게 되면서 설계와 가공을 일원화하려는 요구가 높아지고 있다.

또한 CIM의 기술 부분의 기능적 영역을 그림 2-18과 같이 생각할 때 기술 정보의 관리 통제를 위해 가장 기본적인 단계인 CAD/CAM과 실제 공작 기계군을 제어하기 위한 확장된 개념의 DNC(Distributed Numerical Control) 시스템은 필수 불가결한 요소로 부각되고 있다.

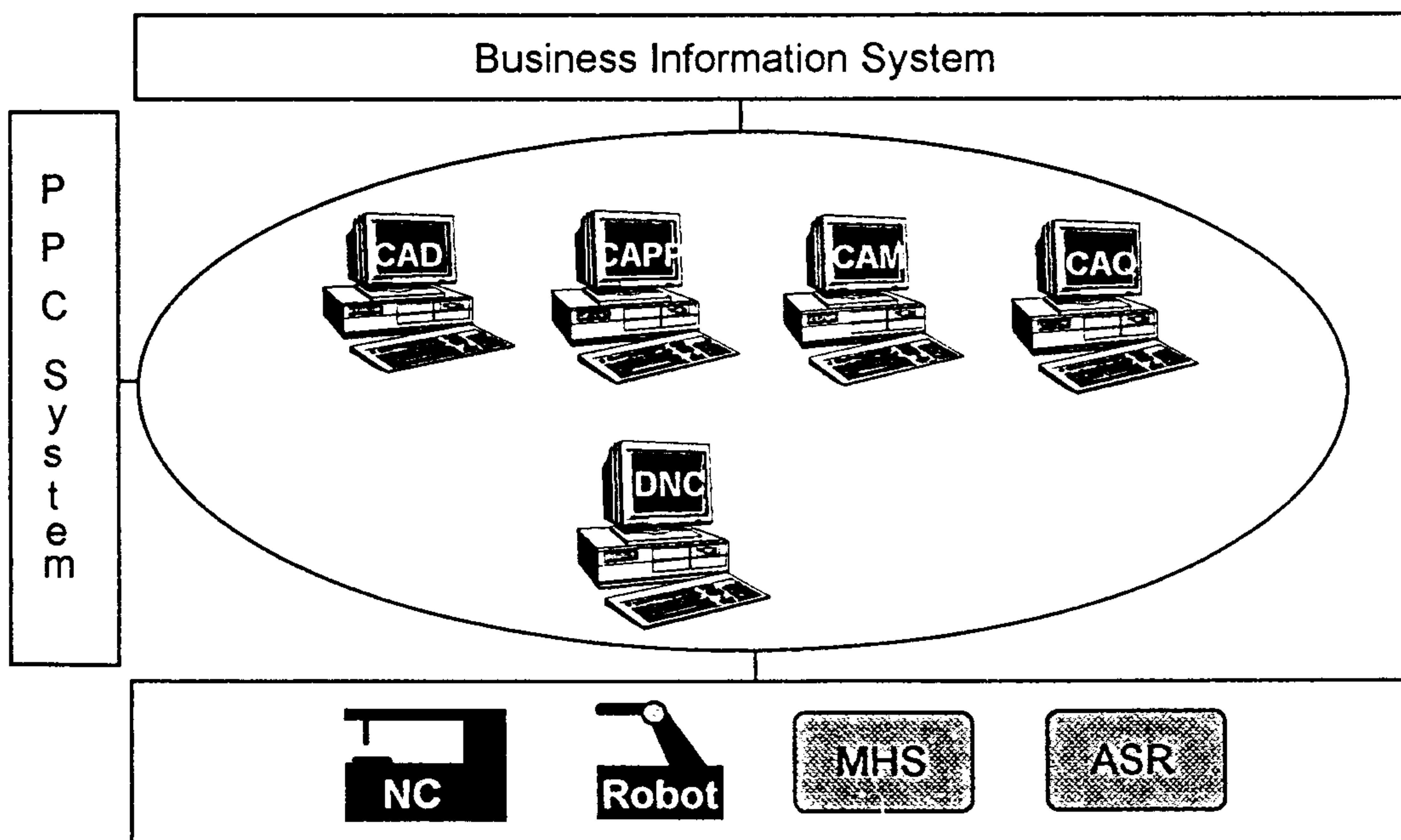


그림 2-18 CIM의 기술 부분의 기능적 영역 구조

이러한 DNC 시스템의 필요성에 따라 국내 기업들도 외국에서 시스템을 도입 운영함으로써 가공기계의 가동율을 높이고 생산성 향상에 경주하고 있으나 가격이 비싸고 수입된 소프트웨어인 관계로 실제 통합을 위한 확장성에 많은 어려움이 따르고 있다.

국내에서도 단위 인터페이스를 위한 BTR(Behind Tape Reader)방식의 DNC(Direct Numerical Control)화에 대한 연구나 셀 레벨에서의 제어등에 관한 연구가 진행되어 왔으나 실제 전체 시스템 내에서 유기적으로 통합된 DNC 시스템에 대한 연구는 미진한 실정이다.

본 연구에서는 NC 공작 기계의 중간 매체인 천공 테이프를 사용하지 않는 Tapeless를 지향하여 CAD/CAM 시스템의 NC 데이터와 제어 정보를 호스트 컴퓨터 혹은 파일 서버에 저장했다가 작업 계획에 따라 NC 공작 기계군의 위성 컴퓨터로 정보를 전송하여 이를 관리 통제하는 계층적, 분산적 개념의 DNC 시스템을 개발하고자 한다.

본 연구에서 DNC 시스템의 기본 구성은 CAD/CAM으로 부터의 정보 및 기타 가공 정보를 저장하는 Bulk Memory에 해당되는 Server Station과 NC 데이터와 기타 기술 정보를 처리하고 각 NC 가공 기계에 전송하는 Host Station 및 이들 NC 데이터를 NC 가공 기계로 Downloading하는 Satellite Station으로 구성하였으며 이들은 계층적 구조로 네트워크에 의해 연결된다. 간략한 구조도는 그림 2-19과 같다.

Server Station은 설계, 가공 및 제반 생산에 관한 기술 정보를 데이터 베이스로 구축하고 있는 파일 서버로써 BOM(Bills of material), CAD 데이터, 작업 스케줄링, NC 데이터 등이 서로 종속적으로 관리되는 공통 데이터 베이스로 구축된다.



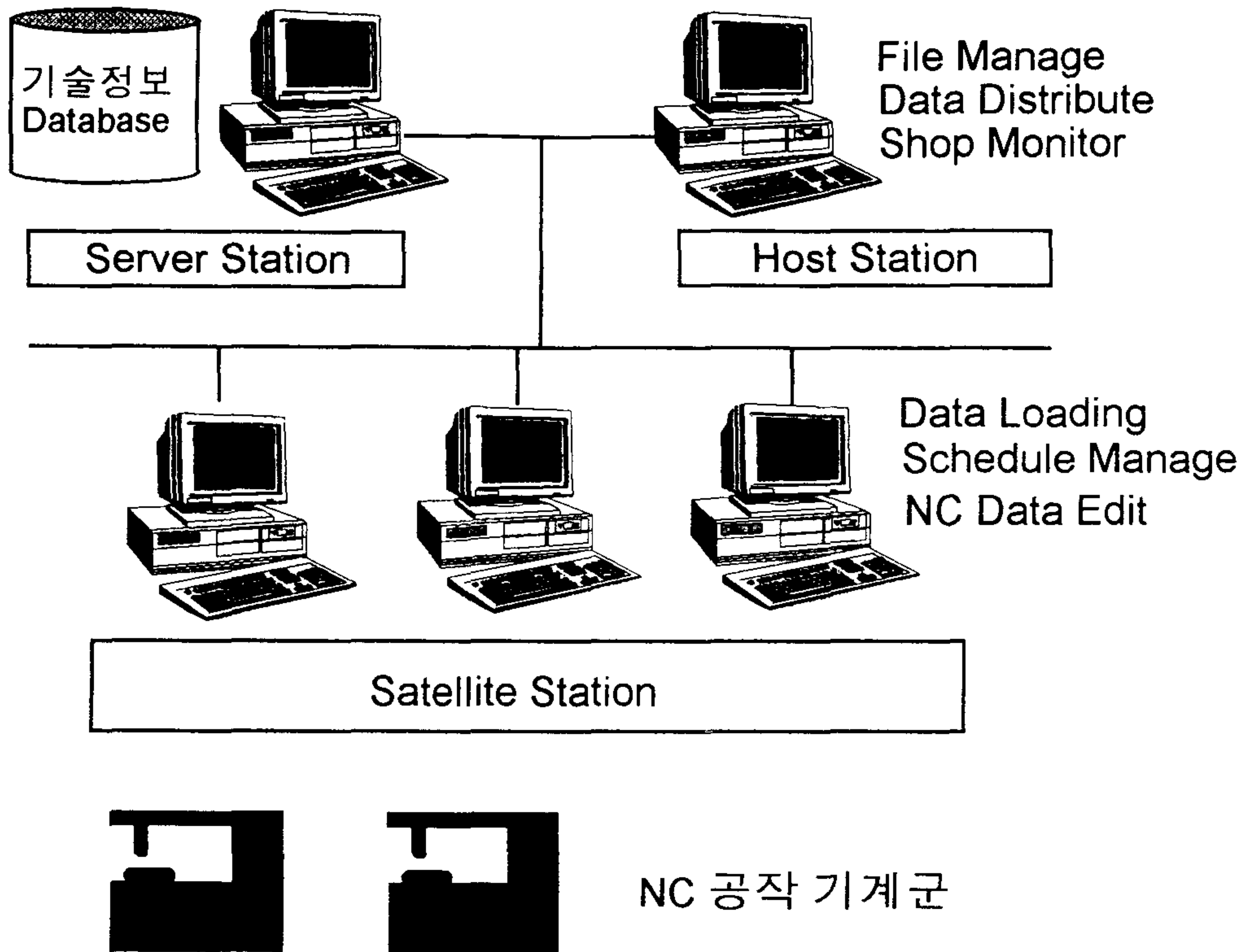


그림 2-19 DNC 시스템 구조도

Host Station은 설계 및 가공을 위한 CAD/CAM 시스템이나 생산 정보나 작업 지시 등을 위한 PPC 시스템과 연결되어 공통 데이터 베이스 내의 파일을 관리하고 데이터를 분배하며 각 기계에 대한 상태를 모니터링함으로써 Satellite Station과 협조 및 제어 역할을 담당한다.

Satellite Station은 공정과 직접 연결되어 공정을 제어한다. 일반적으로 이 Station은 로봇이나 NC 가공 기계등의 생산 설비에 연결되며 바 코드나 테이프 입력용 터미널과 같은 공장 내의 데이터 수집 장치에 연결될 수 있다.

Host Station은 Sun 670을 사용하였으며 19인치 컬러 모니터에 다양한 그래픽 기능과 멀티 윈도우 기능을 제공하는 GUI에 의한 메뉴로 화면 구성을 제공한다. Satellite Station은 널리 보급되어 있는 IBM-PC AT를 사용하여 보수 및 시스템 확장이 유리하도록 구성하였다.

각 Station은 CAD/CAM 시스템이나 기타 시스템들과 함께 LAN에 의해 데이터를 전송하며 Satellite Station과 NC 가공 기계의 접속은 RS-232C 접속을 통한 시리얼 통신을 수행하도록 인터페이스 프로그램을 만들어 중앙 컴퓨터에서 NC 가공 기계에 직접 Data를 전송할 수도 있도록 한다.

CNC 가공 기계에는 통신을 위한 인터페이스가 표준 또는 선택으로 장비되어 있으며 그 대표적인 것으로 데이터를 시리얼로 전송하는 RS-232C 인터페이스가 있다.

RS-232C 인터페이스는 계측용으로 가장 광범위하게 쓰이는 인터페이스의 하나인 GP-IB보다 전송 속도가 느리고 인터페이스에 하나의 단말 장치 밖에 접속할 수 없는 결점도 있으나 접속이 용이하고 데이터 단말 장치와 컴퓨터 간의 케이블 길이는 규격상 15m 까지로 되어 있으나 실제로는 Shield가 있는 Twist pair cable을 사용하면 100m 이상도 가능하다는 장점 등이 있어 많이 사용되고 있다.

EIA-RS-232C 규격에 의하면 2개의 송수신 제어선과 5개의 제어선 및 3개의 그라운드 선이 필요하다. 표 2-4는 EIA-RW-232C 규격에 따른 선의 용도를 보여준다.

표 2-4. EIA-RS-232C

선		용도
1	FG(FG)	Frame Ground(보호용 접지)
2	SD(TxD)	Send Data(송신 데이터)
3	RD(RxD)	Receive Data(수신 데이터)
4	RS(RTS)	Request to Send(송신 요구)
5	CS(CTS)	Clear to Send(송신 허가)
6	DR(DSR)	Data Set Ready(데이터 준비)
7	SG(SG)	Signal Ground(신호용 접지)
8	CD(DCD)	Carrier Detect(수신캐리어 검출)
9	ER(DTR)	Data Terminal Ready(데이터단말 준비)

일반 PC간의 통신에서는 모든 제어선을 사용하지 않아도 최소 3개의 송신 데이터선(SD), 수신 데이터선(RD), 신호용 접지선(SG)이 있으면 통신이 가능하지만 본 연구에서는 NC 데이터 송신측인 컴퓨터와 수신측인 NC 공작 기계와의 접속을 위한 연결은 커넥터 내부에서 다음 그림 2-20와 같이 4, 5, 6, 7, 8, 20핀을 연결한다.

데이터 전송에는 동기 캐릭터를 부가해서 송신하는 동기 방식과 동기 캐릭터를 부가하지 않는 비동기 방식 두가지가 있는데 여기서는 동기 방식을 이용하였으며 C 언어로 간단한 인터페이스 프로그램을 작성하여 이용하였다.

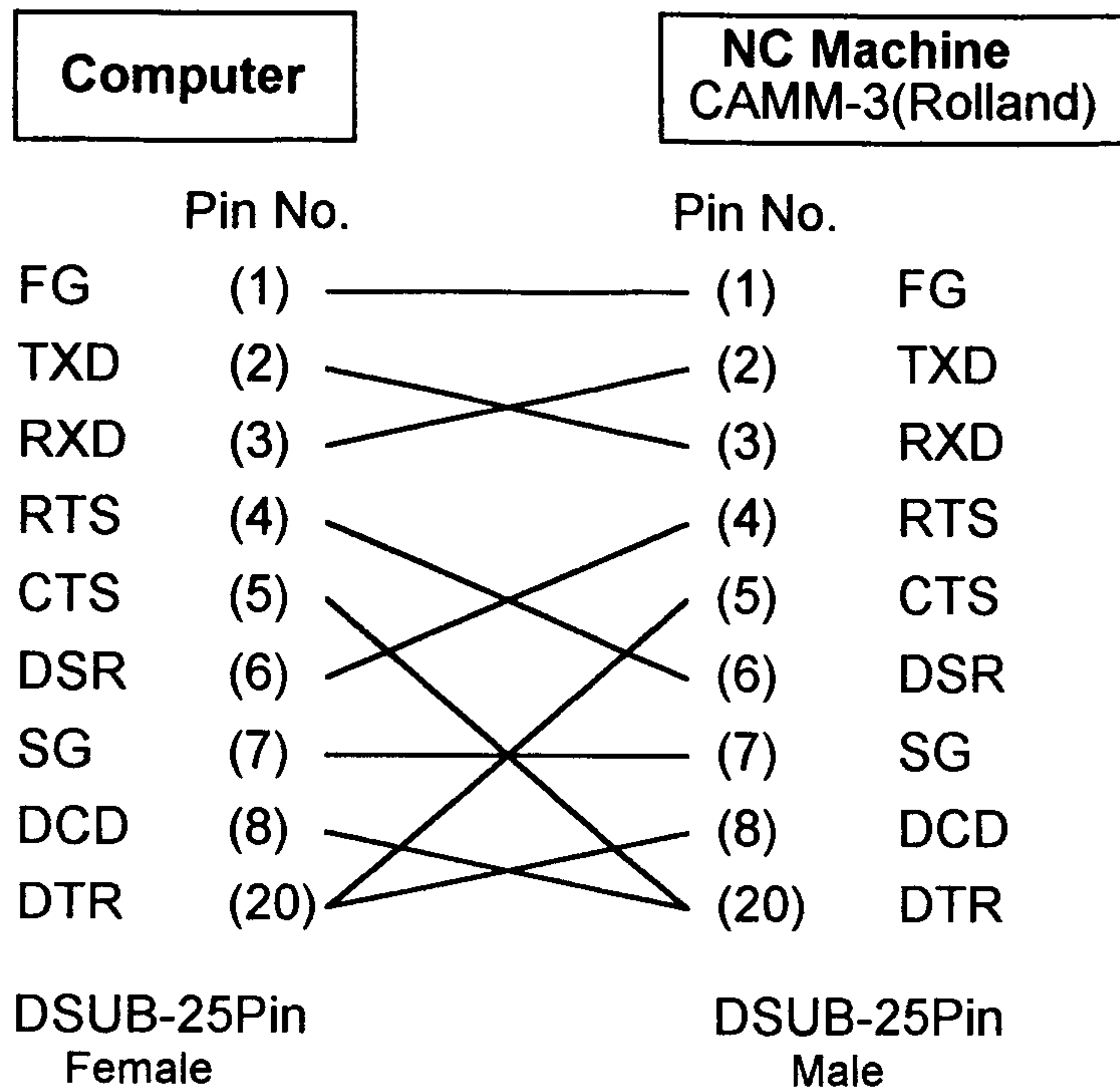


그림 2-20 컴퓨터와 NC간의 RS-232C 핀 접속 예

또한 본 연구에서의 시리얼 연결에 의한 데이터 전송 방식은 다음 표 2-5과 같다.

표 2-5. 데이터 전송 방식

Auto Scale	Off
Z Axis Velocity	Off
Non Buffer Mode	Off
DSR Enable	Off
DCD Enable	Off
Stop Bit	1 bit
Data Bit	8 bit
Parity	Even
Baudrate	1200



본 DNC 시스템은 NC 프로그램 관리 모듈, NC 프로그램 분배 모듈 및 NC 공작 기계 현황 관리 모듈로 크게 구성되어 있으며 각 시스템의 구성을 모듈화하여 수정 및 확장이 용이하도록 구성하고 GUI(Graphic User Interface) 방식으로 메뉴를 구성하여 현장의 이용자로 하여금 대화식으로 쉽게 접근할 수 있도록 하였다.

또한 공작 기계 제어 방식은 중앙 컴퓨터에서 모두 제어할 수도 있고 중앙 컴퓨터의 부하를 줄이기 위해 분산화된 시스템을 구성할 수도 있게 하였다. 실제 현장에 적용은 위성 컴퓨터에 의한 분산화된 시스템을 구성함으로써 중앙 컴퓨터의 고장 시에도 전 시스템이 중단되지 않고 위성 컴퓨터 단위로 진행할 수 있어야 할 것이다.

한편 기계 가동 상태를 알 수 있도록 NC 공작 기계로 부터 기계 현황 데이터를 중앙 컴퓨터에서 받아 기계의 상태를 간단히 모니터링할 수 있도록 상태를 화면 상에서 볼 수 있도록 하였다. 그림 2-21은 본 DNC 시스템의 화면 예를 보여준다.

본 DNC 시스템은 현장 내에서 처리해야 할 NC 프로그램을 효율적으로 관리할 수 있고 또한 신속 정확한 데이터 전달이 가능하여 NC 공작 기계의 가동률을 증대시킬 수 있다.

향후 작업 일정 계획에서의 최적 스케줄링이 연계되고 공작 기계 내의 적응 제어를 위한 기계 감시 제어 모듈이 접속될 수 있으면 완전한 DNC 시스템으로써 구축될 수 있을 것이다.





여 백

## 제 3 장 CIM Prototype 연구



여 백

## 제 3 장 CIM Prototype 연구

### 제1절 개 요

지난 수년 간 CIM은 개념적으로 존재해 왔다. CIM이 생산성의 측면에서 주요한 긍정적인 영향을 가져올 수 있다고 일반적으로 믿고 있으나 오늘날 공장에서 CIM이 구축되어 운영, 유지되는 것이 그리 흔한 일은 아니다.

CIM은 컴퓨터 시스템을 이용하여 설계, 생산 계획 및 스케줄 제조, 판매와 마케팅 등 제조 업체내의 모든 중요한 기능들을 통합한다.

설계자는 제품 생산의 리드 타임을 줄이기 위해 설계 과정 초기에 제조 정보를 필요로 하고 관리자는 생산, 판매 및 수요에 따른 제품을 효율적으로 스케줄하기 위한 판매 및 마케팅 정보를 필요로 하며 영업에서는 리드 타임을 정확히 산출하기 위한 생산정보를 필요로 한다. 이 때 CIM은 이러한 정보를 잘 정의되고 자동화된 방법으로 통합하여 운용될 수 있도록 해야한다.

통합은 CIM 시스템에 있어 가장 중요한 개념이다. 판매, 제조 설계 등의 기업 내의 개별적인 기능들과 공장 내의 자동화 기기들은 꽤 오랫동안 컴퓨터화된 정보 시스템을 사용해 왔지만 서로 독립적으로 개발되었으며 상호 기능들 간의 정보 교환이나 필요한 목적을 갖는 통합은 이룩하지 못했다. 이러한 상황을 일반적으로 자동화의 군도 (islands of automation)라고 일컫는다.

이러한 CIM 시스템의 설계와 개발은 극도로 복잡한 일이며 명확

한 계획과 치밀한 실행이 아니면 완수되기가 힘든 것이다.

CIM 아키텍처는 CIM의 실제적인 제어 소프트웨어나 현장 적용이 아니고 통합 시스템 개발을 위한 기초를 제공하는 전체적인 골격을 뜻하는 것으로 시스템의 Top-Down적인 관점을 제공하며 이는 개개의 서브 시스템이 다른 서브 시스템과 호환성을 갖도록 Bottom-up 구현이 이루어져야 한다.

Anderson 등은 CIM 아키텍처를 통합 제조 시스템을 설계하거나 수정하는데 필요한 기초를 제공하는 원칙이나 기능적인 모델 또는 지침들의 구조적인 상관 관계라고 정의했고 Skevington과 Hsu는 가상적인 시스템 구성의 복합적인 모델이라고 정의했다.

표준 CIM 아키텍처의 개발은 1980년대 초반 부터 연구되기 시작했으며 대다수 일정한 형태의 계층적 모델을 제시하였다. 즉 각 단계에서의 대상이 세분화된 대상으로 갈라져가고 그 각각은 계층구조의 하위 레벨의 상위 대상이 되는 형태다.

최근 일부에서 비계층적인 CIM 모델을 제시했다. 이 모델은 각각의 구성 요소가 임의의 다른 요소를 직접적으로 제어하거나 다른 요소에 의해 제어되지 않는 형태이다.

이에 본 연구에서는 CIM 구축을 위한 제반 사항을 고려하여 CIM 시스템의 prototype을 연구하고 전체 CIM의 셀 레벨 단계에서 CAD/CAM을 위주로 한 prototype을 개발한다.

## 제2절 CIM의 구성 요소 및 구조

### 2.1. CIM 구성 요소

CIM의 출현이 제조 기술과 컴퓨터의 발달에 기인한 바와 같이 CIM에서의 컴퓨터의 역할은 매우 중요하다. 컴퓨터의 역할을 강조한 CIM의 구성 요소로는 CAD(Computer Aided Design), CAPP(Computer Aided Process Planning), CAM(Computer Aided Manufacturing), CAQC(Computer Aided Quality Control), AS/RS(Automated Storage and Retrieval System) 등 기술 부문과 PPC(Production Planning and Control) 시스템으로 대변될 수 있는 생산 관리 부문으로 나누어 생각할 수 있다.

CAD는 부품 및 제품 설계 그리고 공구 및 Jig/Fixture 설계 등에 관한 기능이며, CAPP는 공정 계획에 관한 기능, CAM은 하드웨어, 생산 계획, 가공, 조립 및 정비에 관한 기능을 포함하며 CAQC는 검사 및 품질 관리 기능, AS/RS는 원자재, 반제품 및 완제품의 저장 및 검색에 관한 기능을 포함하며 PPC는 수주에서 출하까지의 생산계획 및 관리 기능을 포함한다.

설계 과정은 제조 활동 전 과정 중에 최초의 공정이며 시작 단계이기 때문에 가능한 한 설계 단계에서 부터 제조 활동 전반에 걸친 문제점을 파악하고 반영해야 한다. 이에 따라서 설계 단계에서 제품 개발에 필요한 각종 시뮬레이션, 가공을 위한 CAM 시스템과의 연계 및 EDB(Engineering Database)와의 연계 등이 고려되어 통합하려고 할 뿐만 아니라 제조 기술 부문이나 심지어 영업 등의 관리 부분과



도 상호 연관을 가지면서 업무가 추진되고 있다.

또한 데이터 수집의 발전이나 DBMS를 통한 정보 통합이 이들 구성 요소를 하나로 통합시키는데 큰 기여를 하고 있다.

CIM의 구성을 업무에 따라 계획과 실행으로 나누어 고려할 때 다음 그림 3-1과 같이 생각할 수 있다.

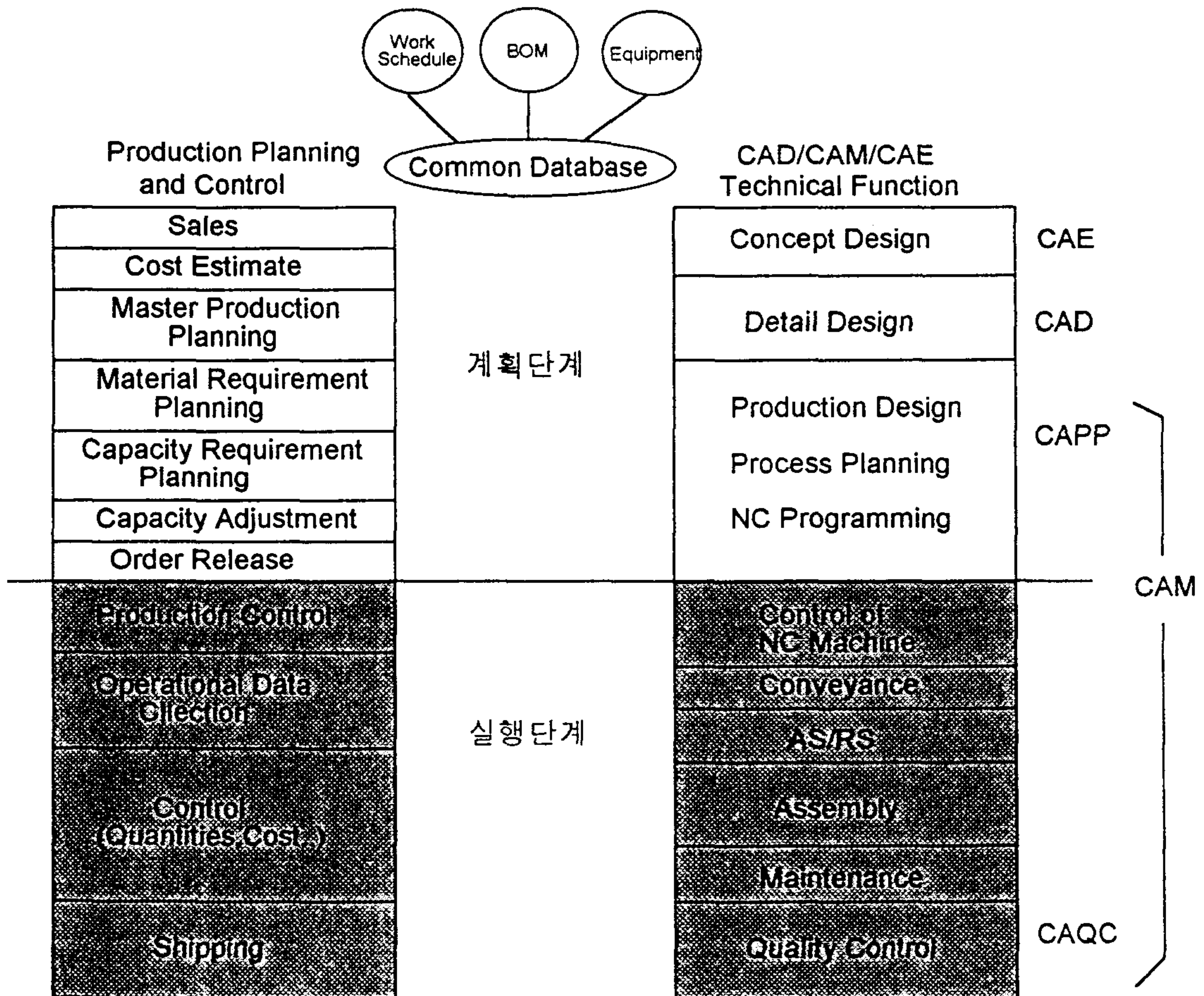


그림 3-1. CIM의 구성 요소

## 2.2. CIM 모델

### 가. NIST의 계층 구조 CIM 모델

CIM의 대표적인 모델중의 하나가 AMRF(Automated Manufacturing Research Facility)의 NIST(Nation Institute for Standards and Technology)에서 개발되었다.

AMRF는 Group Technology 기법을 이용하여 부품군을 만들고 이를 제조하기 위해 설계된 유연 생산 시스템(FMS)을 목적으로 하며 그림 3-2과 같은 계층 제어 개념에 근거하여 상세한 설명을 제공할 수 있다.

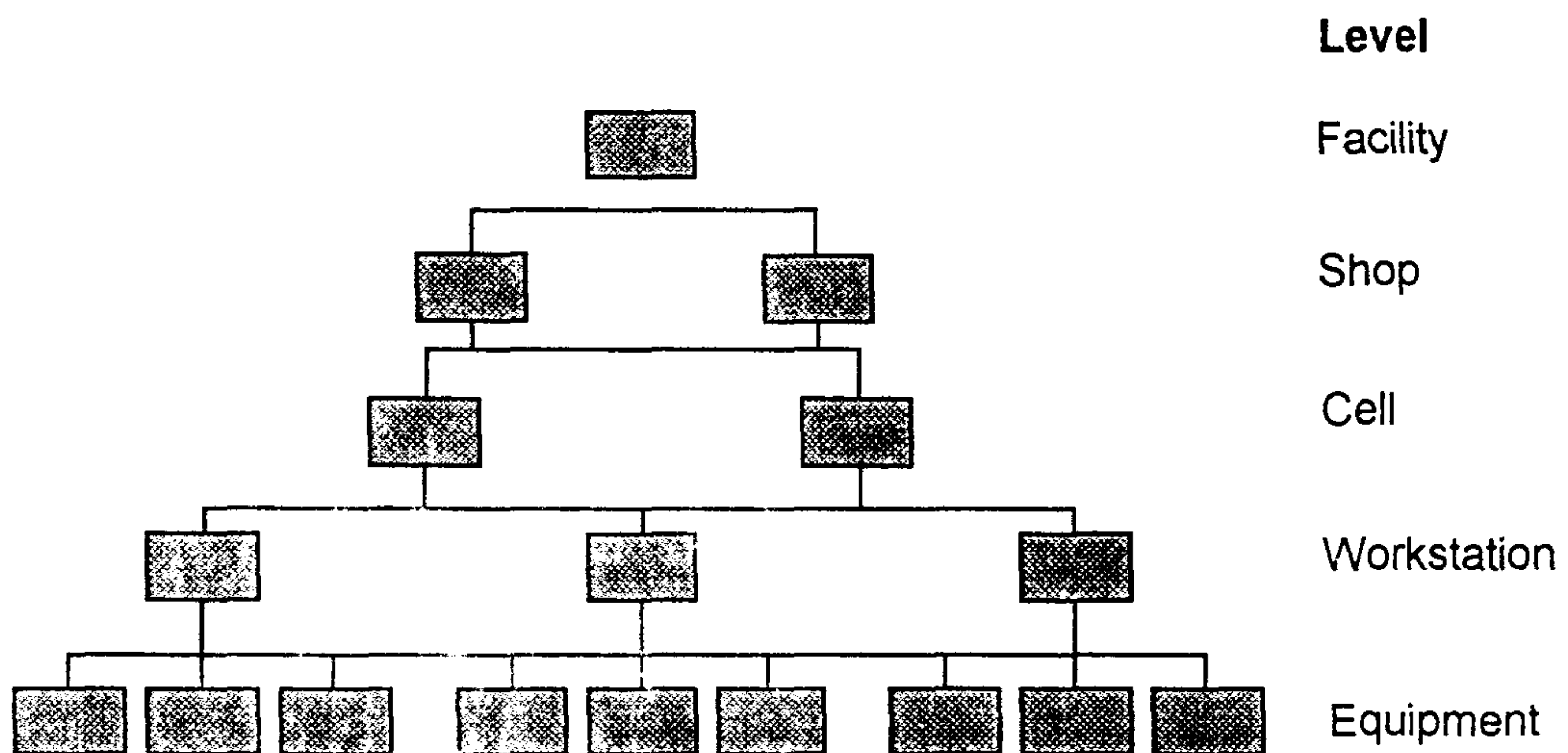


그림 3-2 NIST 5단계 계층 제어 CIM 모델

NIST의 계층 구조는 Facility Level, Shop Level, Cell Level, Workstation Level, Equipment Level의 5단계로 구성된다.

Facility 레벨은 NIST 구조의 최상위 레벨로써 가격, 재고, 노동을



등과 같은 장기적인 기능을 제공한다. Shop 레벨은 생산 Cell들 간의 행위를 조절하고 각 Cell에 필요한 자원을 할당한다. Cell 레벨은 Workstation을 통해 필요한 작업을 할당하고 순서를 결정하여 Workstation의 행위를 감독한다. Workstation 레벨은 Equipment 컨트롤러의 활동 순서를 정하거나 제어한다. 전형적인 Workstation은 로봇이나 공작기계, 자재 보관 창고 등으로 구성될 수 있다. Equipment 레벨의 컨트롤러는 공작 기계나 로봇 앞단의 컴퓨터로써 workstation 컨트롤러로부터 단계적으로 명령을 수신하고 각각의 공작 기계나 로봇에 요구되는 형태로 명령을 변환한다.

#### 나. CIM-OSA

유럽 공동체에서도 CIM 아키텍처를 개발하기 위해 많은 연구가 진행되고 있으며 특히 이를 위해 ESPRIT project 688을 통해 CIM-OSA(Computer Integrated Manufacturing-Open System Architecture)가 결성되었다.

이 CIM-OSA의 목표는 생산 업체에서 각 기능이 요구하는 모든 정보 요구를 지원하는 Open System 아키텍처를 제공하는 것으로써 CIM-OSA Reference Architecture와 CIM-OSA Particular Architecture의 두가지 기준 프레임으로 구성된다.

Jackson과 King은 CIM-OSA 프로젝트로 ADEPT라고 하는 계층적 레퍼런스 아키텍처를 소개했다. 제어 계층은 Factory Level, Department Level, Cell Level의 3단계로 구성되며 이는 NIST 계층의 상위 3레벨(Facility, shop, Cell)에 해당된다. 이 모델의 특징적인 것은 Observer 모듈이 있어 이 모듈이 시스템 구성 요소를 감시하여 데

이터와 각 기능의 완벽을 보장한다.

ADEPT 시스템은 현재 독일 PCB 생산 프랜트에 장치된 Pilot System을 이용해 평가 중이다.

#### 다. 비계층적 제어 모델

Hatvany, Duffie, Piper 등은 앞서 논의한 계층적인 제어 모델과는 다른 비계층적 제어 모델을 제시했다. 비계층적 모델은 여러개의 엔터티들로 구성되고 이들 상호간에 다른 엔터티를 직접적으로 제어하거나 제어되지 않으며 대신 이들 엔터티는 전체적 시스템의 목표를 만족시키기 위해 서로 협동한다. 이는 감독을 위한 의사 결정이 중앙에 있기 보다는 정보의 수집 단계에 있어야 한다는 것이 주요한 동기가 되었다.

부품과 기계를 포함하는 각 엔터티가 모델링될 때 부품 엔터티가 공정을 요구하고 타당한 기계가 유용하면 실행됨으로써 오로지 두 엔터티 사이에서만 동의되고 이를 상위에서 제어하는 엔터티는 포함되지 않는다는 것이다.



### 제3절 CIM Prototype 개발 연구

본 연구에서는 중소 기계 가공 업체의 CIM에 대한 prototype을 제시한다. 이는 NIST 계층 구조 모델을 참고하였고 Sever & Client 방식에 의한 Open System Architecture로 구성하였다.

그림 3-3과 같이 CAD/CAM, DNC, PPC 및 기타 S/W 시스템 (CAPP, CAQC, ...)간의 정보 교환을 제어하기 위한 중앙 시스템을 두고 이 시스템이 관련 시스템에 데이터의 일관성을 유지 시킨다.

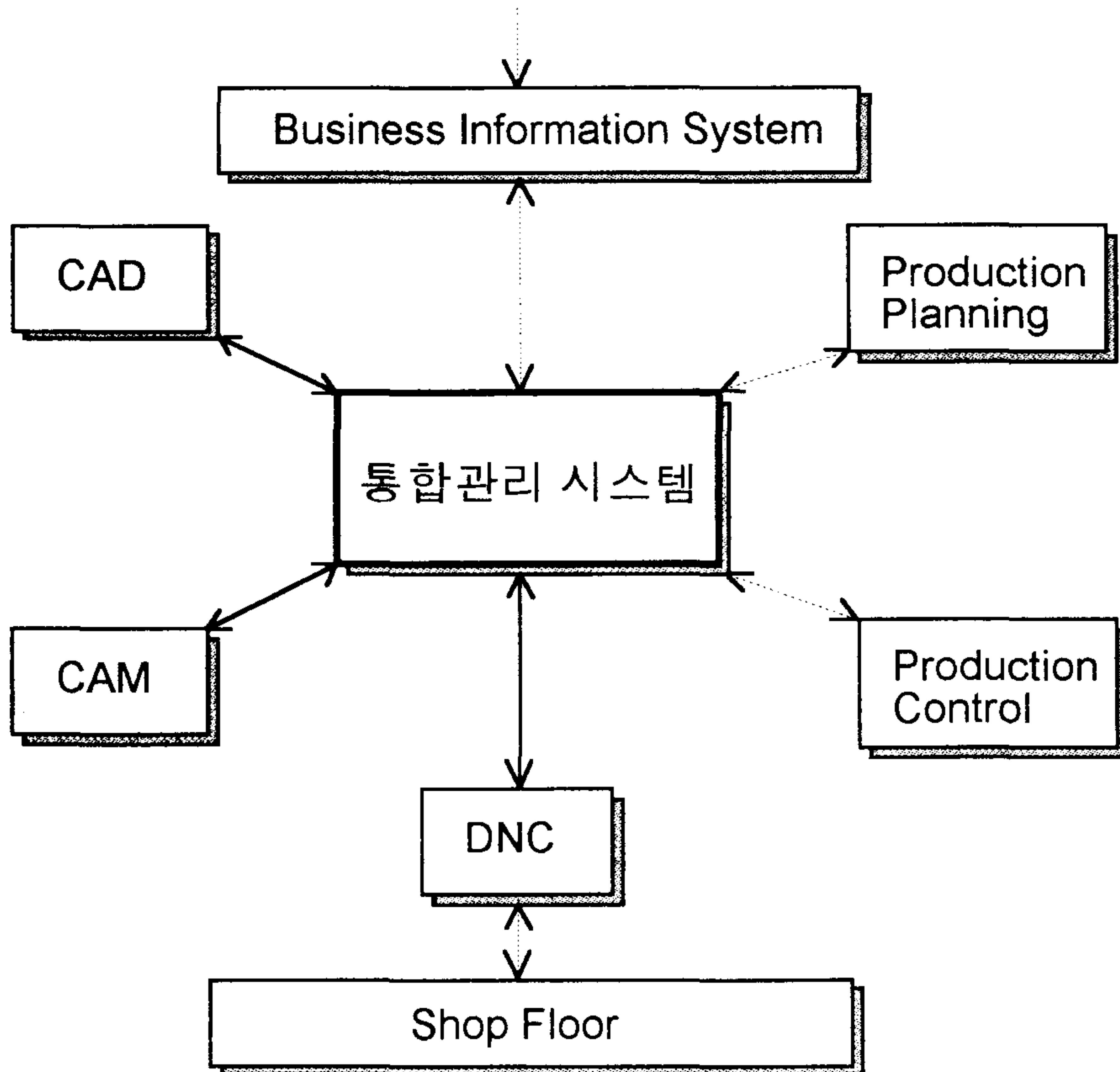


그림 3-3. CIM PROTOTYPE 구조도

상위에서 Business Information System으로 부터 정보를 하위의 설계 및 생산측에서 전달받고 이러한 설계와 가공에 이르는 기술 부문과 수주에서 출하에 이르는 관리 부문이 하나의 통합된 시스템으로 구성된다.

제품 개발 및 설계에서의 CAD/CAM/CAE가 2차원 도면 작업을 위한 일반 PC용 범용 시스템부터 3차원 모델링 및 FEM(Finite Element Method) 해석과 가공을 위한 EWS용의 중형 시스템이 설치되어 개발, 설계, 가공등에 활용된다.

모든 기하학적 형상에 대한 데이터는 중간 파일 형식으로 중앙의 공동 데이터 베이스에서 관리되며 상호 호환할 수 있기 때문에 한 제품에 대한 모델링은 어느곳에서나 모두 쓰일수 있다.

본 연구에서 PC측에서의 CAD 시스템은 일반 범용 CAD인 AutoCAD R.11와 Personal Designer Ver 4.0가 설치되었고, 설계 해석을 위한 FEM 소프트웨어는 MSC/PAL이 설치되어 이용되었다. 또한 가공을 위해서 Personal Machinist Ver 2.0이 CAM 소프트웨어로 활용되었다.

CAD에서 만들어지는 Partlist, 부품정보 및 기타 기술 정보와 설계 해석 결과에 대한 기술 정보나 가공 데이터 등이 NFS를 통해 호스트 컴퓨터의 공통 데이터 베이스 내에 저장되며 이는 PPC등의 다른 부분에서 활용될 수 있다.

이들 정보들은 기준 정보 관리 모듈에서 BOM, 작업 스케줄 및 설비 등에 필요한 정보들과 함께 관리된다. 이들 기준 정보는 MPS (Master Production Scheduling), MRP(Material Requirement Planning), CRP(Capacity Requirement Planing) 등의 모듈로 생산 계획

수립을 위해 요구된다.

생산 계획이 수립되면 스케줄된 작업 지시가 NC 프로그램과 함께 DNC 시스템에 전송되고 DNC 시스템에서 각 공작기계의 PC로 NC 프로그램을 Download하여 원하는 부품을 가공하게 된다.

실제 가공을 위해 간단한 Modelling Machine인 Roland사의 CAMM-3 Model PNC-3000을 밀링용으로 이용하였으며 PC나 SUN W/S과 RS-232C Interface를 통해 데이터를 Downloading할 수 있도록 하였다.

이러한 통합화된 일련의 시스템이 다음 그림 3-4에서 보여준다.

여기서 통합화는 정보의 통합화와 생산 물류의 통합화를 위한 CIM 내의 구성 성분에 대한 통합으로써 데이터의 공유성, 데이터 접근의 용이성 및 즉시성이 매우 중요하다.

실제 적용을 위해서는 각각의 업무, 기능 간의 인터페이스와 정보 계층 구조 구축이 선결 과제가 되어야 하며 이를 위한 데이터 베이스 구조와 네트워크 작업이 이루어져야만 할 것이다.

본 연구에서의 전체 네트워크의 기본은 Local Network로써 산업용 LAN(Norm IEEE 802.3) 표준이고 Topology는 Linear Bus형태의 EtherNET을 이용하였으며 프로토콜은 TCP/IP이다.

PPC에 관련된 부분은 본 연구단에서 진행 중인 과제들으로써 향후 CAD/CAM에 관련된 본 시스템과 연계될 예정이다.

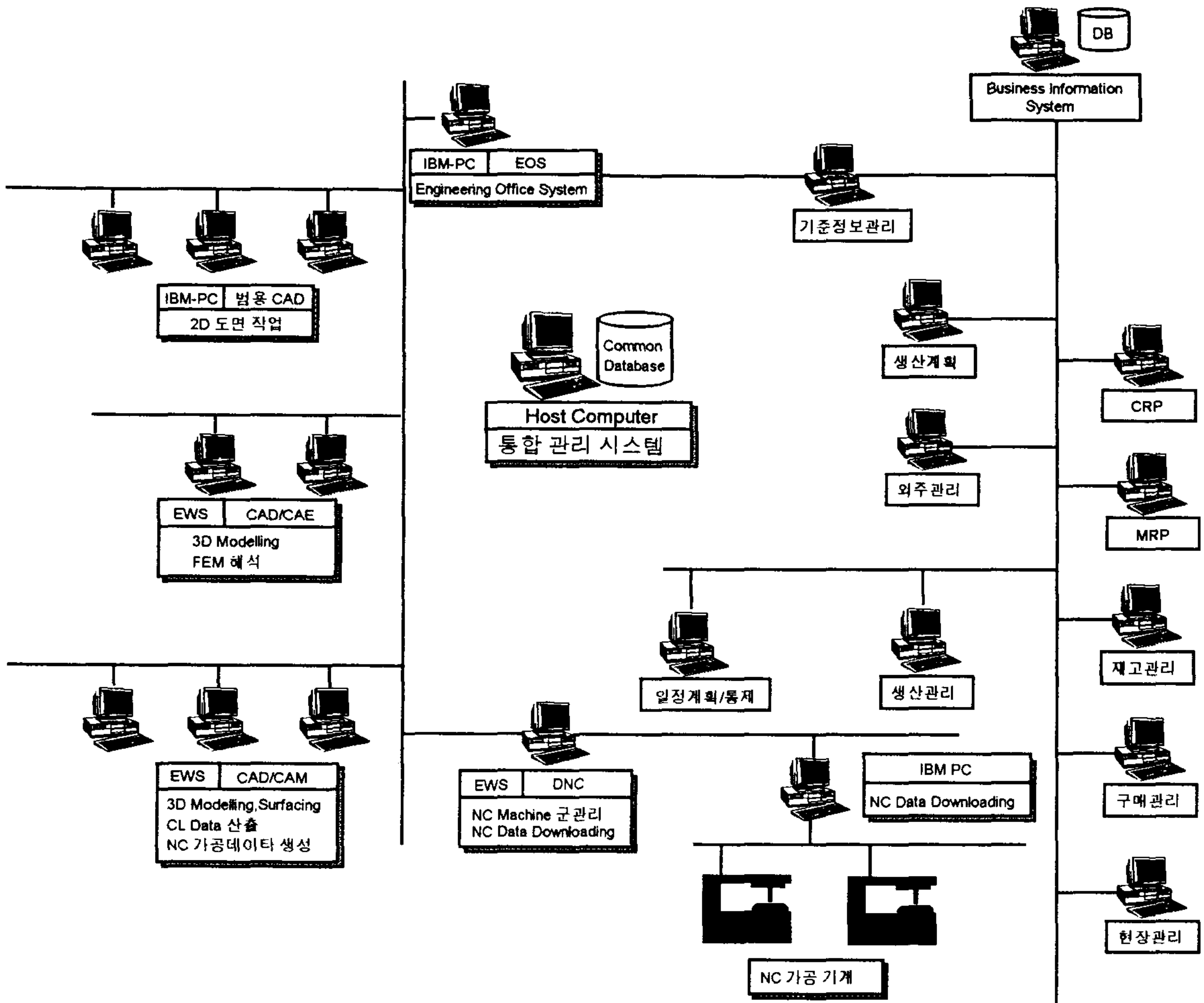


그림 3-4 CIM 시스템 구성도



여 백

## 제 4 장 결 론

여 백

## 제 4 장 결 론

### 제1절 연구 결과

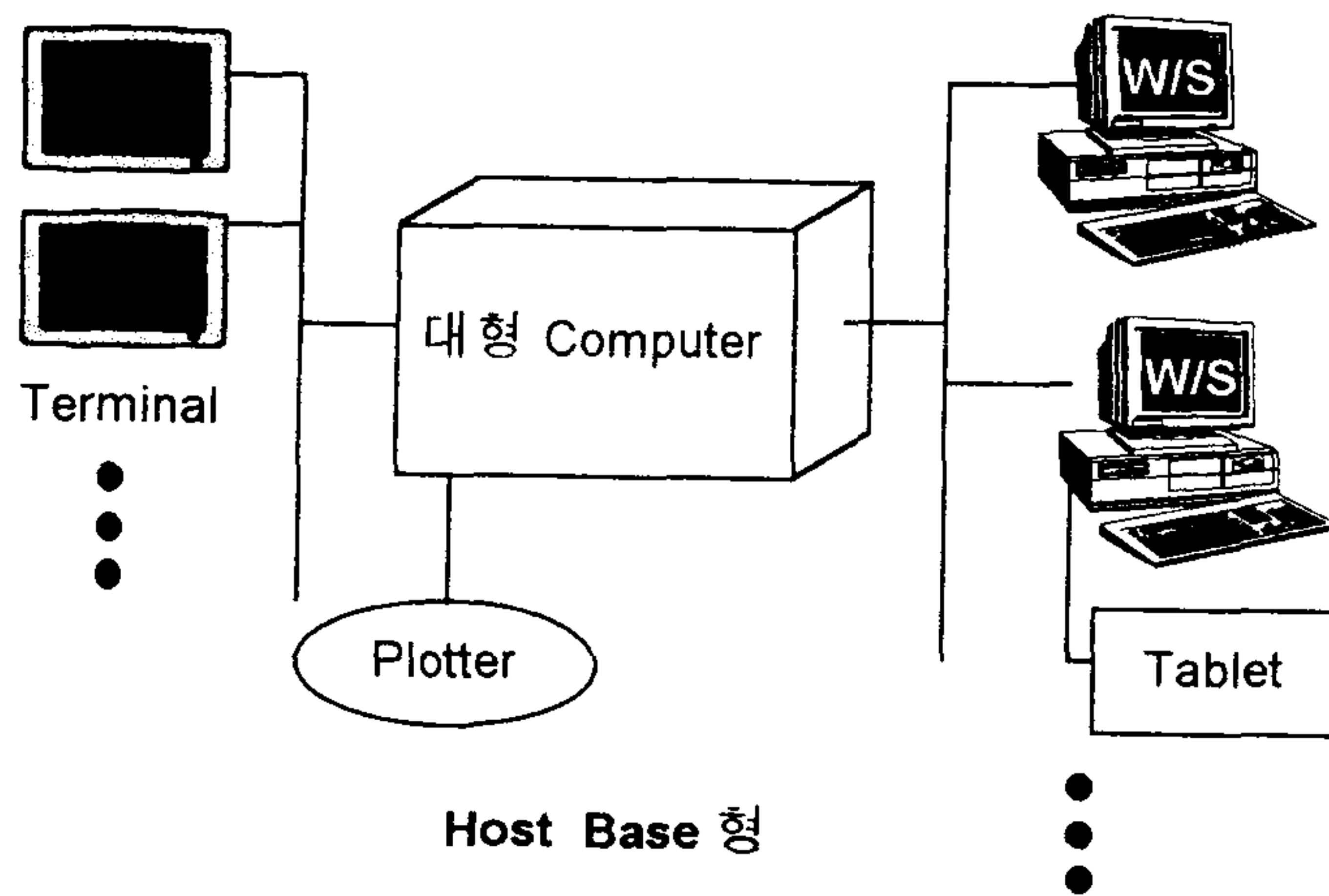
설계 및 생산 기간의 단축, 설계 품질의 향상, 표준화 촉진, 제품의 품질 향상, 다품종 소량 생산에의 대처 등 설계 및 가공에서의 성력화에 의한 생산성 향상을 위해 1980년 중반 이후로 국내에 CAD/CAM 시스템의 도입이 급격히 증가되고 있으나 CAD/CAM 시스템은 그 자체만으로 효과를 기대할 수 있는 것이 아니며 시스템의 통합적 운용이나 응용 소프트웨어의 개발등 난제가 놓여 있는 게 사실이다.

이에 전 년도에 금형 설계 전용 CAD 시스템 개발로 설계의 효율화를 높였고 이를 가공과 연계하기 위해 CAD 정보로 부터 간단한 2차원 NC 데이터를 산출함으로써 설계 및 가공을 일원화 시키고자 하였다. 이는 CAD 시스템 자체적인 활용으로써 매우 효율적이나 전체의 통합된 형태의 구성은 미비하였으므로 본 연구를 통해 CAD/CAM 시스템의 통합 운용에 대한 지침을 제시하고자 하였다.

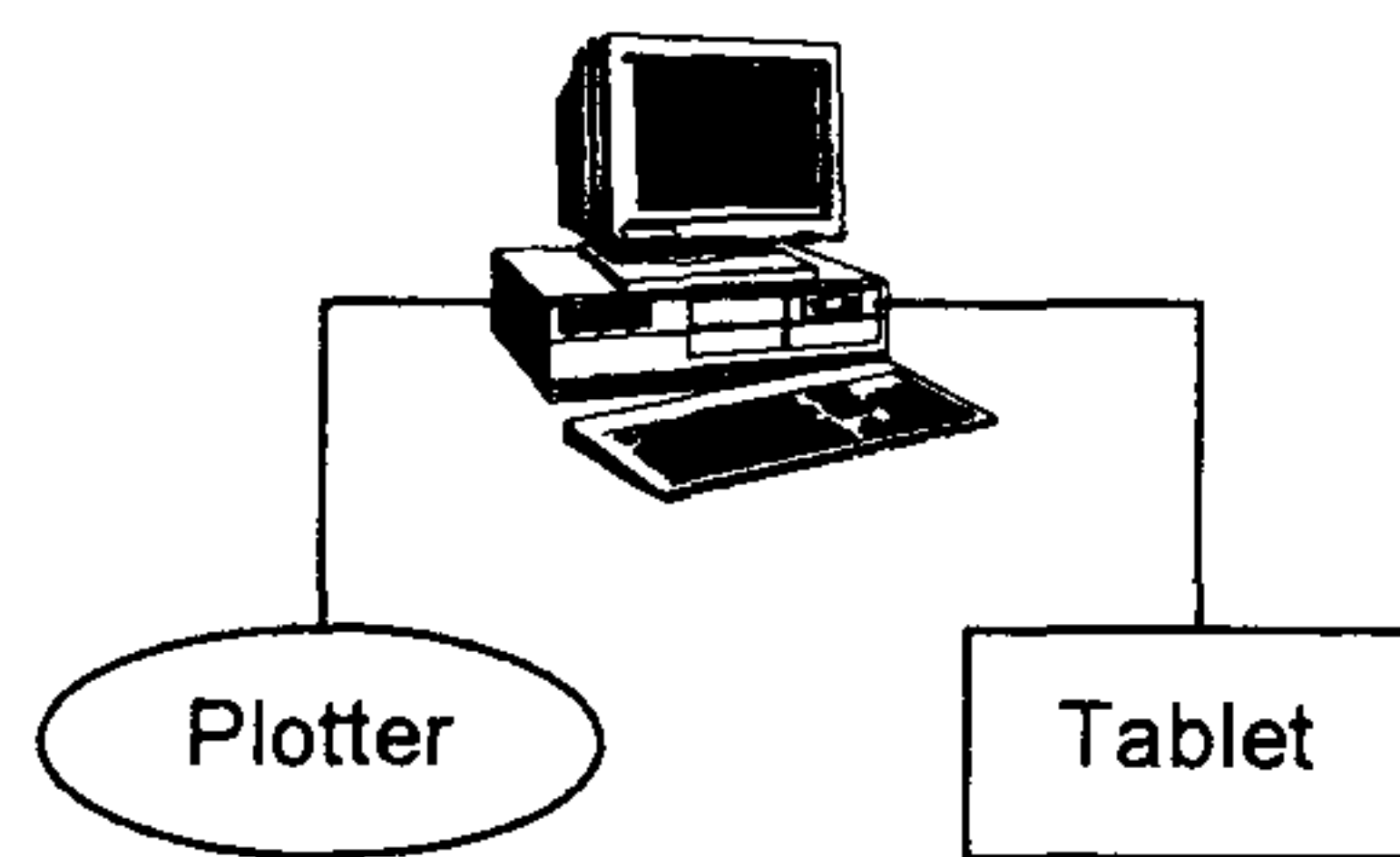
CAD/CAM 시스템은 그 운용에 있어서 업무 특성이나 운영 요원의 전문화 정도 등에 따라 개방형, 폐쇄형 또는 이들의 혼합형 시스템 등으로 운용될 수 있고 그 이용 형태는 Host-base형, Stand-alone형, Network형 등으로 나뉘볼 수 있다. 앞으로의 추세로 볼 때 시스템 운영과 지원을 위한 별도의 운영실을 두고 각 부서에 CAD/CAM실을 설치 운영하는 혼합형 시스템의 운용 형태와 Server & Client 방식의 LAN에 의한 Network형의 이용 형태가 CAD/CAM



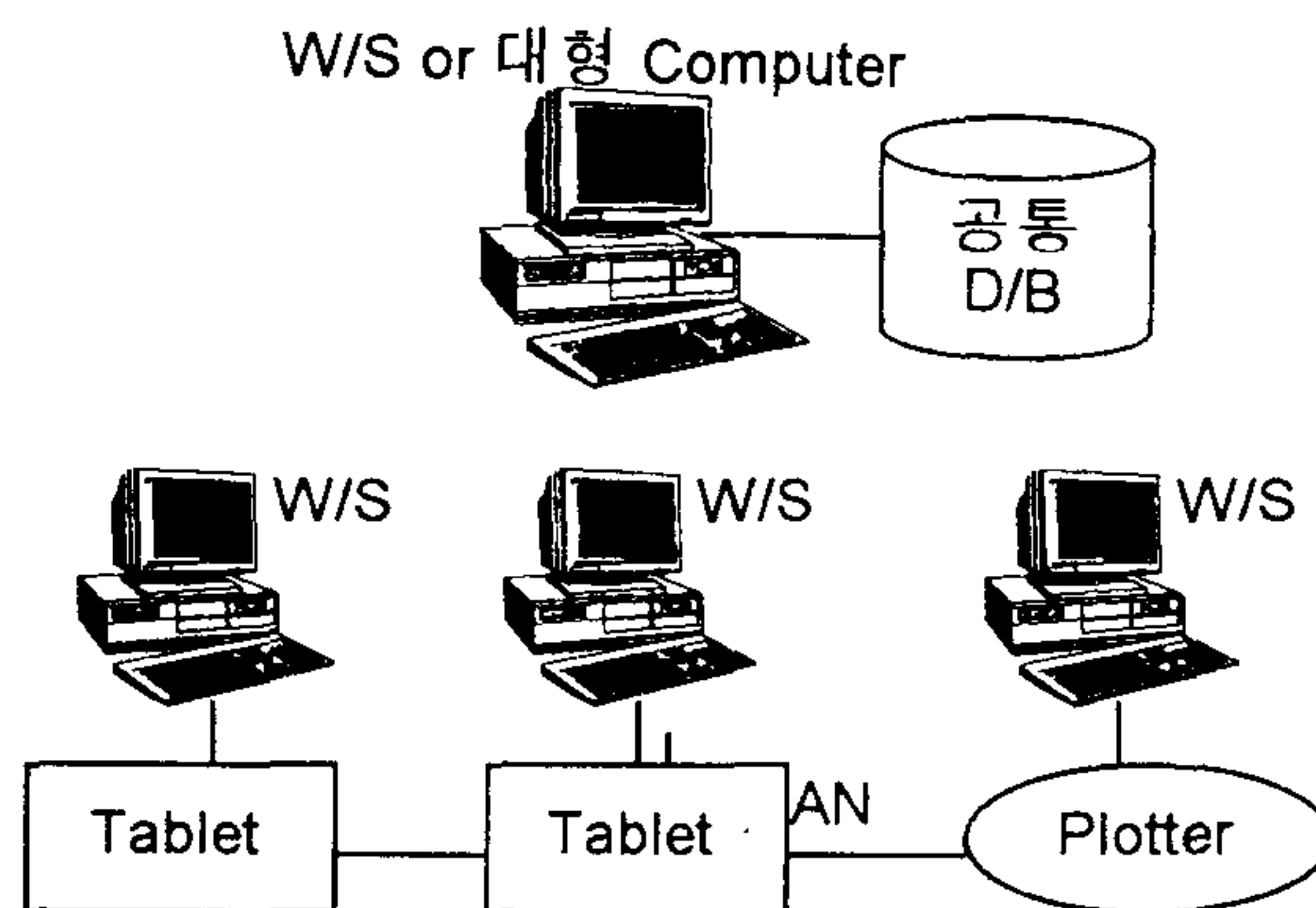
시스템의 주류를 이룰 것으로 예상된다. 그림 3-5는 CAD/CAM 시스템의 각각의 이용 형태를 보여준다.



Host Base 형



Stand-alone 형



Network 형

그림 3-5. CAD/CAM 시스템의 이용 형태

따라서 본 연구에서는 이러한 Network형으로 운용되는 CAD/CAM 시스템의 통합 운용을 위해 다양한 시스템 간의 도면 데이터 관리, 설계에서의 기준 정보 데이터 및 생산에 필요한 정보 데이터 관리 등 기본적인 통합 관리 시스템이 개발되었다.

이를 위해 CAD/CAM 네트워킹이 이루어졌고 이중 CAD 시스템간의 데이터 호환을 위해 중간 파일의 형식이 연구되었다. 네트워킹은 LAN을 이용 모든 시스템들을 Linear Bus 방식으로 연결하고 Open System으로 Server & Client 방식에 의해 구성하였으며 NFS를 이용하여 공통 데이터 베이스를 구축하였다.

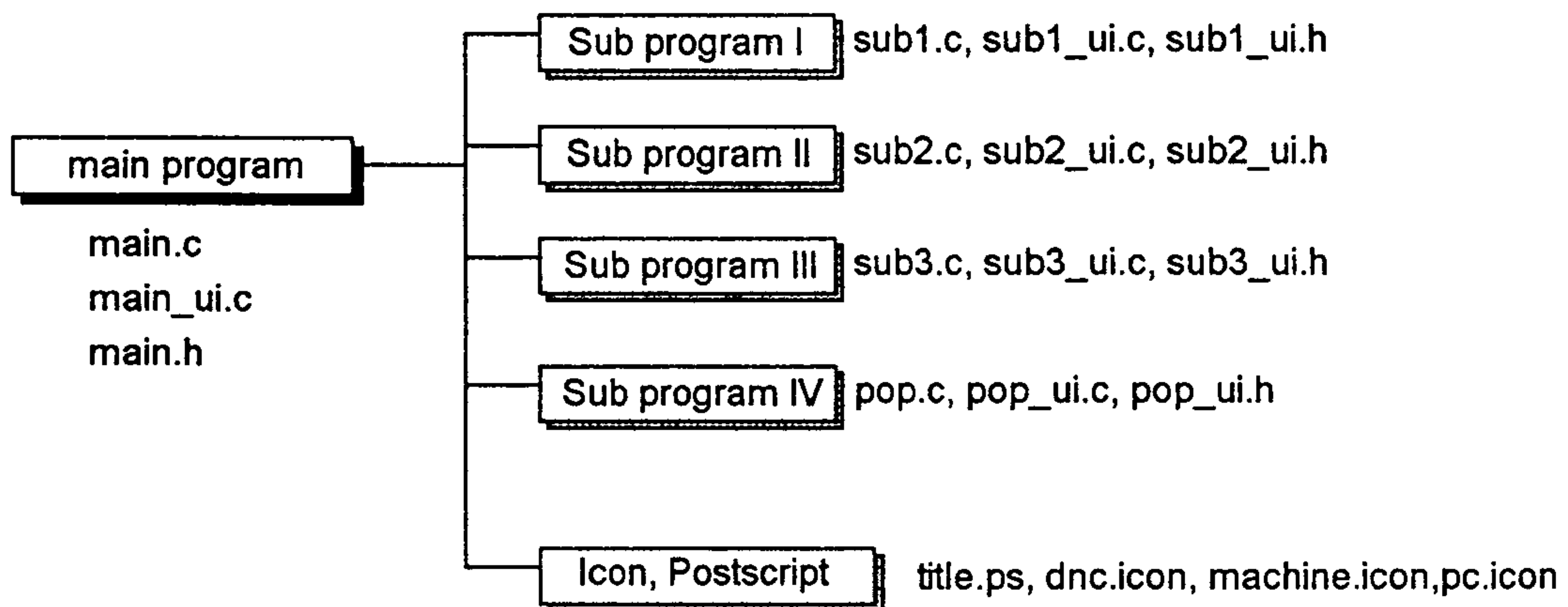
기하학적 형상에 대한 정보는 IGES와 DXF 파일 포맷의 중간 파일 형식을 검토하고 2D 도면 작업이나 와이어 프레임 모델은 DXF를 이용하고 3차원 모델링 정보는 IGES를 이용하여 CAD 소프트웨어로 연계될 수 있도록 하였다.

이와 함께 부품 정보나 기술정보 등 설계로부터의 기준 정보를 생성시키고 이를 중앙에서 효율적으로 관리할 수 있는 통합 관리 시스템이 Openwindow 하에서 그래픽과 GUI에 의해 개발되었다.

또한 이러한 네트워킹과 통합 관리에 의해 NC 프로그램 및 기타 가공 정보를 기존의 천공 테이프 방식이 아닌 CAD/CAM 시스템에 의한 Tapeless 가공을 지향하여 설계로부터 가공에 대한 시뮬레이션을 거쳐 NC 데이터를 생성, 이를 해당 기계에 직접 downloading 하여 가공할 수 있도록 DNC 시스템을 개발하였다. 이를 위해 CAM 시스템으로 부터의 CL 데이터를 후처리하기 위한 포스트 프로세서가 개발되었고 이로 부터 각 기계에 전송은 RS-232C 인터페이스를 이용하였다.

이들 프로그램은 하나의 메인 프로그램을 포함한 5개의 소스 프로그램, 5개의 GUI 인터페이스 프로그램과 그에 따른 각각의 헤더 파일 및 그래픽 아이콘 파일과 postscript 프로그램 등으로 구성되었다. 또한 PC측에서는 기준 정보 생성 프로그램이나 DNC의 Satellite Control측의 프로그램이 본 시스템에 포함되어 있다.

부록에 소스 프로그램의 일부를 수록하였고 그림 3-6는 통합 운용 관리를 위한 호스트 측의 프로그램 구성도이다.



Sub program I : 설계 도면 및 기준 정보 통합 관리 프로그램

Sub program II : 가공 데이터 관리 운용 프로그램

Sub program III : DNC 프로그램

Sub program IV : 그래픽 인터페이스를 위한 아이콘 파일  
또는 postscript 프로그램

그림 3-6 개발 시스템의 프로그램 구성도

본 시스템은 설계와 가공의 일원화를 위한 CAD/CAM의 통합 운용 시스템으로써 본 연구에서는 이를 통해 CIM의 Prototype을 제시하고자 하였다.

CIM의 구성 요소를 살펴보고 이들을 통합하기 위한 기존의 CIM의 모델을 검토하고 이를 참조로 CAD/CAM을 포함한 설계에서 생산에 이르는 기술 부문과 수주에서 출하에 이르는 관리 부문을 통합하기 위한 CIM Prototype을 개발 제시하였다.

이는 설계로 부터 공정 계획, 생산 계획 및 통제, 작업 지시, 가공, 조립에 이르기 까지 통합 데이터를 공유하고 이들은 수주와 출하에 관련된 생산을 위한 제반 관리 부문과 연계될 수 있는 시스템으로 구성할 수 있도록 하였으며 이는 본 연구에서 개발된 CAD/CAM 통합 운용 시스템을 확장함으로써 쉽게 이루어질 수 있도록 하였다.



## 제2절 기대 효과

Stand-alone형이 주류를 이루고 있는 국내 CAD/CAM 시스템은 비교적 고가의 장비로써 주로 CAM을 지향하고 있기 때문에 국내 생산 업계에서 설계 제도를 위해 저가의 CAD 시스템을 이용하고 이와는 별도의 시스템으로 CAM 시스템을 운영하는 실정으로 설계에서 가공에 이르는 과정을 일원화 및 통합화하는데 어려움이 있다.

이에 본 연구에서는 중형 CAD/CAM 시스템과 PC CAD 시스템을 통신 회선으로 연결하여 이를 통합 관리할 수 있는 시스템을 구축하였다.

이를 통해 종래 수작업으로 이루어 지던 설계 제도를 쉽게 접근할 수 있는 PC용 이차원 범용 CAD 시스템으로 개발된 전용 설계 시스템을 이용하여 쉽게 작업할 수 있으며 어느 정도의 도형 편집과 도형 제도등 개개의 정보를 처리하고 공유함과 동시에 이들이 CAD/CAM 시스템 및 기준 정보 관리 부문과 통합 관리 운용됨으로써 CAD 설계의 성력화와 표준화 촉진 및 공수 절감 등으로 생산성을 높일 수 있을 것이다.

또한 셀 단위 자동화 단계에서 통합 생산 시스템을 구축하기 위한 가공 기계의 통합 관리는 매우 중요하며 이를 위해 NC 공작기계를 군관리 하기 위한 DNC 시스템을 개발하였다.

이는 기존의 테이프에 의한 가공을 벗어나 Tapeless와 온라인에 의한 NC 데이터의 공급으로 NC 가공 기계의 가동률을 향상시키고 현장 내의 기술 정보를 효율적으로 관리할 수 있을 것이다.

또한 본 시스템은 CIM을 위한 시스템으로써 이를 통해 CIM Prototype 개발 연구로 국내 제조업의 CIM 구축에 레퍼런스 모델을

제시하였다. 이는 CAD/CAM 시스템의 도입과 함께 CIM을 구축하기 위한 국내 기업에 지침이 될 수 있을 것이다.

본 연구의 기술적, 경제적 측면의 기대 효과를 요약하면 다음과 같다.

- 설계 및 생산의 성력화
- 표준화 촉진
- 설계 및 생산 현장 내의 기술 정보의 효율적 관리
- NC 가공 기계의 가동률 향상
- 설계와 생산의 Line 일원화에 의한 생산성 향상
- 통합된 시스템 구축에 의한 공수 절감 및 납기 단축
- CIM 구축을 위한 기본적 요소 기술 개발
- 컴퓨터 통합 생산 시스템으로의 도약

여 백

# 부 록



여 백

## 메인 프로그램

```

#include <stdio.h>
#include <sys/param.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/dir.h>
#include <xview/xview.h>
#include <xview/panel.h>
#include <xview/font.h>
#include <xview/xv_xrect.h>
#include <xview/font.h>
#include <xview/notice.h>
#include <xview/textsw.h>
#include <gdd.h>
#include <xvps/pscanvas.h>

#include "main.h"
#include "main_ui.h"
#include "sub1_ui.h"
#include "sub2_ui.h"
#include "sub3_ui.h"
#include "pop_ui.h"

#include "title.h"

sub1_dwin_objects      *sub1_dwin;
sub2_dncwin_objects   *sub2_dncwin;
sub3_ttywin_objects   *sub3_ttywin;
pop_popup_objects     *pop_popup;

extern void    del_field();

/*
 * Instance XV_KEY_DATA key.  An instance is a set of related
 * user interface objects.  A pointer to an object's instance
 * is stored under this key in every object.  This must be a
 * global variable.
 */
Attr_attribute  INSTANCE;
NeWStoken news;

void
main(argc, argv)
    int      argc;
    char     **argv;
{
    main_main_window_objects      *main_main_window;
    Xv_Font      font;

    /*
     * Initialize XView.
     */
    xv_init(XV_INIT_ARGC_PTR_ARGV, &argc, argv, XV_USE_LOCALE, TRUE, 0)
    INSTANCE = xv_unique_key();

    /*
     * Initialize user interface components.
     */
    main_main_window = main_main_window_objects_initialize(NULL, NULL);
    /*
     * Canvas is to convert News canvas
     */
    news=(NeWStoken) xv_get(main_main_window->title_canvas,
                           PSCANVAS_NEWSTOKEN);

    sub1_dwin = sub1_dwin_objects_initialize(NULL,
                                              main_main_window->main_window);
    sub2_dncwin = sub2_dncwin_objects_initialize(NULL,

```

```

        main_main_window->main_window);
sub3_ttywin = sub3_ttywin_objects_initialize(NULL,
        main_main_window->main_window);
pop_popup = pop_popup_objects_initialize(NULL,
        main_main_window->main_window);

xv_set(sub1_dwin->dwin, FRAME_SHOW_HEADER, FALSE, NULL);
xv_set(sub2_dncwin->dncwin, FRAME_SHOW_HEADER, FALSE, NULL);
xv_set(sub3_ttywin->ttywin, FRAME_SHOW_HEADER, FALSE, NULL);
xv_set(pop_popup->popup, FRAME_SHOW_HEADER, FALSE, NULL);

/*
 * Turn control over to XView.
 */
xv_main_loop(main_main_window->main_window);
exit(0);
}

/*
 * return an allocated char * that points to the last item in a path.
 */
char *
getfilename(path)
char *path;
{
    char *p;

    if (p = rindex(path, '/'))
        p++;
    else
        p = path;
    return strcpy(malloc(strlen(p)+1), p);
}

void
close_win()
{
    if(xv_get(sub1_dwin->dwin, XV_SHOW) == TRUE) {
        if(xv_get(sub1_dwin->dwin, FRAME_CMD_PUSHPIN_IN) == TRUE)
            xv_set(sub1_dwin->dwin,
                FRAME_CMD_PUSHPIN_IN, FALSE,
                XV_SHOW, FALSE, NULL);
        else
            xv_set(sub1_dwin->dwin, XV_SHOW, FALSE, NULL);
    }
    if(xv_get(sub2_dncwin->dncwin, XV_SHOW) == TRUE) {
        if(xv_get(sub2_dncwin->dncwin, FRAME_CMD_PUSHPIN_IN) == TRUE)
            xv_set(sub2_dncwin->dncwin,
                FRAME_CMD_PUSHPIN_IN, FALSE,
                XV_SHOW, FALSE, NULL);
        else
            xv_set(sub2_dncwin->dncwin, XV_SHOW, FALSE, NULL);
    }
    if(xv_get(sub3_ttywin->ttywin, XV_SHOW) == TRUE) {
        if(xv_get(sub3_ttywin->ttywin, FRAME_CMD_PUSHPIN_IN) == TRUE)
            xv_set(sub3_ttywin->ttywin,
                FRAME_CMD_PUSHPIN_IN, FALSE,
                XV_SHOW, FALSE, NULL);
        else
            xv_set(sub3_ttywin->ttywin, XV_SHOW, FALSE, NULL);
    }
}

int search_file(value)
int value;

```



```

{
    DIR                *dirp;
    struct direct      *dp;
    struct stat        s_buf;
    char               buf[MAXPATHLEN];

    int                cnt = 0;

    if (stat(ROOTDIR, &s_buf) == -1 (s_buf.st_mode & S_IREAD))
        printf("WnPath and s_buffer is Error");
    if (s_buf.st_mode & S_IFDIR) {
        if (!(dirp = opendir(ROOTDIR)))
            printf("WnOpendir is Error");

        while (dp = readdir(dirp))
            if (strcmp(dp->d_name, ".") && strcmp(dp->d_name, "..")) {
                (void) sprintf(buf, "%s%s", ROOTDIR, dp->d_name);
                if(value == 1)
                    xv_set(sub1_dwin->product_list,
                            PANEL_LIST_STRING, cnt++, getfilename(dp->d_name)
                            NULL);
                else if(value == 2)
                    xv_set(sub2_dncwin->product_list,
                            PANEL_LIST_STRING, cnt++, getfilename(dp->d_name)
                            NULL);
                else
                    return 0;
            }
        }
    closedir(dirp);

    return cnt;
}

void del_flist()
{
    int    Cfile_drw, Cfile_nc, i;

    Cfile_drw = xv_get(sub1_dwin->filelist, PANEL_LIST_NROWS);
    Cfile_nc = xv_get(sub2_dncwin->filelist, PANEL_LIST_NROWS);

    if(Cfile_drw != 0) {
        for( i = Cfile_drw-1 ; i >= 0 ; i-- )
            xv_set(sub1_dwin->filelist, PANEL_LIST_DELETE, i, NULL);
    }

    if(Cfile_nc != 0) {
        for( i = Cfile_nc-1 ; i >= 0 ; i-- )
            xv_set(sub2_dncwin->filelist, PANEL_LIST_DELETE, i, NULL);
    }
}

void del_plist()
{
    int    Cfile_drw, Cfile_nc, i;

    Cfile_drw = xv_get(sub1_dwin->product_list, PANEL_LIST_NROWS);
    Cfile_nc = xv_get(sub2_dncwin->product_list, PANEL_LIST_NROWS);

    if(Cfile_drw != 0) {
        for( i = Cfile_drw-1 ; i >= 0 ; i-- )
            xv_set(sub1_dwin->product_list, PANEL_LIST_DELETE, i, NULL);
    }

    if(Cfile_nc != 0) {
        for( i = Cfile_nc-1 ; i >= 0 ; i-- )

```

```

        xv_set(sub2_dncwin->product_list, PANEL_LIST_DELETE, i, NULL);
    }
}

/*
 * Notify callback function for `drw_nc_det'.
 */
void
main_set_proc(item, value, event)
    Panel_item    item;
    unsigned int  value;
    Event         *event;
{
    main_main_window_objects    *ip = (main_main_window_objects *) xv_get(item, XV
ATA, INSTANCE);

    int    i;

    del_flist();

    if(value == 0x0) {
        notice_prompt(ip->main_control, NULL,
            NOTICE_FOCUS_XY,      event_x(event), event_y(event),
            NOTICE_MESSAGE_STRINGS, "이 Button은 이용되지 않습니다", NULL,
            NOTICE_BUTTON_YES,    "확인",
            NULL);
    }

    else if(value == 1) {
        close_win();
        xv_set(sub1_dwin->dwin,
            XV_X,    0,
            XV_Y,    113,
            FRAME_CMD_PUSHPIN_IN, TRUE,
            XV_SHOW, TRUE, NULL);

        i = search_file(1);
        del_field();
    }
    else if(value == 2) {
        close_win();
        xv_set(sub2_dncwin->dncwin,
            XV_X,    0,
            XV_Y,    113,
            FRAME_CMD_PUSHPIN_IN, TRUE,
            XV_SHOW, TRUE, NULL);

        i = search_file(2);
        if(xv_get(sub2_dncwin->file_editor, TEXTSW_FILE) != NULL)
            xv_set(sub2_dncwin->file_editor,
                TEXTISW_FILE, NULLFILE, NULL);
    }
    else if(value == 3) {
        close_win();
        xv_set(sub3_ttywin->ttywin,
            XV_X,    0,
            XV_Y,    113,
            FRAME_CMD_PUSHPIN_IN, TRUE,
            XV_SHOW, TRUE, NULL);
    }
    else if(value == 4) {
        close_win();
    }
    else
        fprintf(stderr, "%nError !");
}

```

```

}

/*
 * Notify callback function for `quit_button'.
 */
void
quit_proc(item, event)
    Panel_item    item;
    Event         *event;
{
    main_main_window_objects    *ip = (main_main_window_objects *) xv_get(item, XV
ATA, INSTANCE);

    xv_set(ip->main_window, FRAME_NO_CONFIRM, FALSE, NULL);
    xv_destroy_safe(ip->main_window);
}

/*
 * Repaint callback function for `title_canvas'.
 */
void
title_proc(pscanvas, newstoken, display, xid, rects)
    PScanvas      pscanvas;
    NewStoken     newstoken;
    Display       *display;
    Window        xid;
    Xv_xrectlist  *rects;
{
    main_main_window_objects    *ip = (main_main_window_objects *) xv_get(pscanvas
EY_DATA, INSTANCE);

    xv_set(pscanvas, PSCANVAS_CLIPRECTS, rects);
    pscanvas_flip();
    PS_fillcanvas("skyblue");
    ps_findfont("Bembo-Bold");
    ps_scalefont(115);
    ps_setfont();

    Define_Rundisplay1("Total",
        "l exch sub .9 l hsbcolor setcolor");
    Rundisplay();

    Define_Rundisplay2("CAD/CAM Manager",
        "l exch sub .9 l hsbcolor setcolor");
    Rundisplay();

    ps_findfont("Bembo-Bold"); ps_scalefont(80); ps_setfont();

    Define_Rundisplay3("KIST/SERI", "l exch sub .0 l hsbcolor setcolor");
    Rundisplay();

    Define_Rundisplay4("By Kim Hyun", "l exch sub .0 l hsbcolor setcolor");
    Rundisplay();

    ps_flush_PostScript();
}

```

```

#include <stdio.h>
#include <sys/param.h>
#include <sys/types.h>
#include <xview/xview.h>
#include <xview/canvas.h>
#include <xview/icon_load.h>
#include <xview/panel.h>
#include <xview/scrollbar.h>
#include <xview/svimage.h>
#include <xview/termsw.h>
#include <xview/text.h>
#include <xview/tty.h>
#include <xview/xv_xrect.h>
#include <xvps/pscanvas.h>
#include <gcm.h>
#include "main_ui.h"

Xv_opaque
main_menu1_create(ip, owner)
    caddr_t      *ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(XV_NULL, MENU_COMMAND_MENU,
                   XV_KEY_DATA, INSTANCE, ip,
                   MENU_GEN_PIN_WINDOW, (Xv_opaque) ip[0], "",
                   NULL);
    return obj;
}

/*
 * Initialize an instance of object `main_window'.
 */
main_main_window_objects *
main_main_window_objects_initialize(ip, owner)
    main_main_window_objects *ip;
    Xv_opaque    owner;
{
    if (!ip && !(ip = (main_main_window_objects *) calloc(1, sizeof (main_main_window_c
s))))
        return (main_main_window_objects *) NULL;
    if (!ip->main_window)
        ip->main_window = main_main_window_main_window_create(ip, owner);
    if (!ip->main_control)
        ip->main_control = main_main_window_main_control_create(ip, ip->main_window);
    if (!ip->msg1)
        ip->msg1 = main_main_window_msg1_create(ip, ip->main_control);
    if (!ip->msg2)
        ip->msg2 = main_main_window_msg2_create(ip, ip->main_control);
    if (!ip->main_setting)
        ip->main_setting = main_main_window_main_setting_create(ip, ip->main_contro);
    if (!ip->quit_button)
        ip->quit_button = main_main_window_quit_button_create(ip, ip->main_control);
    if (!ip->msg3)
        ip->msg3 = main_main_window_msg3_create(ip, ip->main_control);
    if (!ip->title_canvas)
        ip->title_canvas = main_main_window_title_canvas_create(ip, ip->main_window);
    return ip;
}

/*
 * Create object `main_window' in the specified instance.
 */
Xv_opaque
main_main_window_main_window_create(ip, owner)

```

```

        caddr_t      ip;
        Xv_opaque    owner;
    }

    Xv_opaque      obj;

    obj = xv_create(owner, FRAME,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_WIDTH, 1140,
                    XV_HEIGHT, 840,
                    XV_LABEL, "<< Hyun's CAD/CAM Management System for CIM >>",
                    FRAME_CLOSED, FALSE,
                    FRAME_SHOW_FOOTER, TRUE,
                    FRAME_SHOW_RESIZE_CORNER, FALSE,
                    NULL);
    gcm_initialize_colors(obj, "Black", NULL);
    return obj;
}

/*
 * Create object `main_control' in the specified instance.
 */
Xv_opaque
main_main_window_main_control_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque      obj;

    obj = xv_create(owner, PANEL,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 0,
                    XV_Y, 0,
                    XV_WIDTH, WIN_EXTEND_TO_EDGE,
                    XV_HEIGHT, 80,
                    WIN_BORDER, TRUE,
                    NULL);
    gcm_initialize_colors(obj, "Steel Blue", NULL);
    return obj;
}

/*
 * Create object `msg1' in the specified instance.
 */
Xv_opaque
main_main_window_msg1_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque      obj;
    Xv_opaque      msg1_image;
    static unsigned short msg1_bits[] = {
#include "kist_logo.icon"
    };

    msg1_image = xv_create(XV_NULL, SERVER_IMAGE,
                           SERVER_IMAGE_BITS, msg1_bits,
                           SERVER_IMAGE_DEPTH, 1,
                           XV_WIDTH, 64,
                           XV_HEIGHT, 64,
                           NULL);
    obj = xv_create(owner, PANEL_MESSAGE,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 16,
                    XV_Y, 8,
                    XV_WIDTH, 64,
                    XV_HEIGHT, 64,

```



```

        PANEL_ITEM_COLOR, gcm_color_index("Blue"),
        PANEL_LABEL_IMAGE, msg1_image,
        PANEL_LABEL_BOLD, TRUE,
        NULL);
    return obj;
}

/*
 * Create object `msg2' in the specified instance.
 */
Xv_opaque
main_main_window_msg2_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_MESSAGE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 96,
        XV_Y, 16,
        XV_WIDTH, 120,
        XV_HEIGHT, 16,
        PANEL_ITEM_COLOR, gcm_color_index("White"),
        PANEL_LABEL_STRING, "시스템공학연구소",
        PANEL_LABEL_BOLD, TRUE,
        NULL);
    return obj;
}

/*
 * Create object `main_setting' in the specified instance.
 */
Xv_opaque
main_main_window_main_setting_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern int    main_set_proc();
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_CHOICE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 300,
        XV_Y, 16,
        XV_WIDTH, 581,
        XV_HEIGHT, 46,
        PANEL_VALUE_X, 300,
        PANEL_VALUE_Y, 36,
        PANEL_CHOICE_NROWS, 1,
        PANEL_LAYOUT, PANEL_VERTICAL,
        PANEL_ITEM_COLOR, gcm_color_index("Yellow"),
        PANEL_LABEL_STRING, "원하는 Button을 누르시오",
        PANEL_NOTIFY_PROC, main_set_proc,
        PANEL_CHOICE_STRINGS,
        "",
        " CAD/CAM Data 관리 시스템 ",
        " DNC System ",
        " 단말기 Mode ",
        " CAD/CAM SYSTEM ",
        0,
        PANEL_CHOICE_COLOR, 1, gcm_color_index("Red")
        PANEL_CHOICE_COLOR, 2, gcm_color_index("red")
        PANEL_CHOICE_COLOR, 3, gcm_color_index("red")
        PANEL_CHOICE_COLOR, 4, gcm_color_index("Red")
        NULL);
}

```

```

        return obj;
    }

    /*
     * Create object `quit_button' in the specified instance.
     */
    Xv_opaque
    main_main_window_quit_button_create(ip, owner)
        caddr_t      ip;
        Xv_opaque    owner;
    {
        extern void      quit_proc();
        Xv_opaque      obj;

        obj = xv_create(owner, PANEL_BUTTON,
            XV_KEY_DATA, INSTANCE, ip,
            XV_X, 1036,
            XV_Y, 20,
            XV_WIDTH, 64,
            XV_HEIGHT, 21,
            PANEL_ITEM_COLOR, gcm_color_index("Blue"),
            PANEL_LABEL_STRING, " Quit ",
            PANEL_NOTIFY_PROC, quit_proc,
            NULL);
        return obj;
    }

    /*
     * Create object `msg3' in the specified instance.
     */
    Xv_opaque
    main_main_window_msg3_create(ip, owner)
        caddr_t      ip;
        Xv_opaque    owner;
    {
        Xv_opaque      obj;

        obj = xv_create(owner, PANEL_MESSAGE,
            XV_KEY_DATA, INSTANCE, ip,
            XV_X, 96,
            XV_Y, 44,
            XV_WIDTH, 134,
            XV_HEIGHT, 16,
            PANEL_ITEM_COLOR, gcm_color_index("White"),
            PANEL_LABEL_STRING, " 자동화시스템연구단 ",
            PANEL_LABEL_BOLD, TRUE,
            NULL);
        return obj;
    }

    /*
     * Create object `title_canvas' in the specified instance.
     */
    Xv_opaque
    main_main_window_title_canvas_create(ip, owner)
        caddr_t      ip;
        Xv_opaque    owner;
    {
        extern void      title_proc();
        Xv_opaque      obj;

        obj = xv_create(owner, PSCANVAS,
            XV_KEY_DATA, INSTANCE, ip,
            XV_X, 0,
            XV_Y, 80,

```

```
        XV_WIDTH, WIN_EXTEND_TO_EDGE,  
        XV_HEIGHT, WIN_EXTEND_TO_EDGE,  
        PSCANVAS_REPAINT_PROC, title_proc,  
        XV_VISUAL_CLASS, StaticColor,  
        NULL);  
    return obj;  
}
```

## 기타 프로그램

(기존 정보 통합 통합 운용관리 및 DNC )

```

#include <stdio.h>
#include <strings.h>
#include <errno.h>
#include <sys/param.h>
#include <sys/stat.h>
#include <sys/dir.h>
#include <sys/types.h>
#include <sys/fcntl.h>
#include <sys/termios.h>
#include <xview/xview.h>
#include <xview/panel.h>
#include <xview/textsw.h>
#include <xview/xv_xrect.h>
#include <xview/notice.h>
#include <gdd.h>
#include <xview/notify.h>

#include "main.h"
#include "sub1_ui.h"
#include "sub2_ui.h"
#include "pop_ui.h";

sub1_dwin_objects      *sub1_dwin;
sub2_dncwin_objects   *sub2_dncwin;
pop_popup_objects     *pop_popup;

extern char*   getfilename();
extern int     search_file();
extern void    del_flist();
extern void    del_plist();

FILE          *fp;

char          p_name[BUFSIZE];          /* Product Name      */
char          det_cadm[BUFSIZE];       /* DRW DIR or NC DIR */
char          file_name[BUFSIZE];      /* *.drw 이냐 *.dat냐 */
char          directory[BUFSIZE];      /* 선택된 Directory  */
int           flist_option = 0;        /* Sorting Option    */

char          current_value[BUFSIZE]; /* 현재 선택된 Filename */

#ifdef MAIN
/*
 * Instance XV_KEY_DATA key. An instance is a set of related
 * user interface objects. A pointer to an object's instance
 * is stored under this key in every object. This must be a
 * global variable.
 */
Attr_attribute INSTANCE;

void
main(argc, argv)
    int          argc;
    char         **argv;
{
    sub1_dwin_objects      *sub1_dwin;

    /*
     * Initialize XView.
     */
    xv_init(XV_INIT_ARGC_PTR_ARGV, &argc, argv,
            XV_USE_LOCALE, TRUE,
            0);
    INSTANCE = xv_unique_key();

    /*
     * Initialize user interface components.

```



```

        */
        subl_dwin = subl_dwin_objects_initialize(NULL, NULL);

        /*
        * Turn control over to XView.
        */
        xv_main_loop(subl_dwin->dwin);
        exit(0);
    }

#endif

/*
 * Notify callback function for `drwmng_set'.
 */
void
flist_set_proc(item, value, event)
    Panel_item    item;
    int           value;
    Event         *event;
{
    subl_dwin_objects    *ip = (subl_dwin_objects *) xv_get(item, XV_KEY_DATA, INST

    subl_dwin_objects    *subl_dwin;

    del_flist();

    if(value == 0) {
        strcpy(det_cadm, DRWDIR);
        strcpy(file_name, DRWFILE);
        xv_set(ip->drw_nc_msg,
            PANEL_LABEL_STRING, "도면 파일이 선택되고 관리됩니다",
            PANEL_LABEL_BOLD, TRUE,
            NULL);
    }
    else {
        strcpy(det_cadm, NCDIR);
        strcpy(file_name, NCFILE);
        xv_set(ip->drw_nc_msg,
            PANEL_LABEL_STRING, "NC DATA 파일이 선택되고 관리됩니다",
            PANEL_LABEL_BOLD, TRUE,
            NULL);
    }
}

/*
 * Get the file list
 * /usr3/users/hkim/flist PROGRAM을 돌린다.
 * 성공하면 1을 return하고 실패하면 0을 return
 */
int file_list()
{
    static char    cmd_buff[BUFSIZE];

    if( !det_cadm[0] ) strcpy(det_cadm, DRWDIR);
    if( !file_name[0] ) strcpy(file_name, DRWFILE);

    sprintf(directory, "%s%s", ROOTDIR, p_name, det_cadm);
    sprintf(cmd_buff, "%s %s %s %ld",
        SHELLPRG , directory , file_name , flist_option);

    if( system(cmd_buff) < 0 ) {
        perror("execl");
    }
}

```

```

        return 0;
    }

    system("");

    return 1;
}

/*
 * "~/flist" PROGRAM에 의해 나온 "~/temp/outfile" FILE을 열고
 * 그 내용을 list화면에 뿌린다.
 */

void
flist_display(value)
int    value;
{
    char    STREAM[BUFSIZE];

    int    Count_pre, Count, i;

    Count_pre = xv_get(sub1_dwin->filelist, PANEL_LIST_NROWS);

    if ((fp = fopen(OUTFILE, "r")) == NULL) {
        perror(OUTFILE);
        exit(0);
    }

    Count = 0;
    while (fgets(STREAM, BUFSIZE, fp) != NULL) {
        if(value == 1)
            xv_set(sub1_dwin->filelist,
                PANEL_LIST_STRING, Count++, STREAM, NULL);
        else
            xv_set(sub2_dncwin->filelist,
                PANEL_LIST_STRING, Count++, STREAM, NULL);
    }
    if(Count_pre > Count) {
        for(i = Count_pre-1 ; i >= Count ; i--) {
            if(value == 1)
                xv_set(sub1_dwin->filelist, PANEL_LIST_DELETE, i, NULL);
            else
                xv_set(sub2_dncwin->filelist, PANEL_LIST_DELETE, i, NULL);
        }
    }
}

/*.....
PANEL_LIST_DELETE 부분을 이렇게 고친다.
if(Count_product > Count_pre) {
    del_row_no = Count_product - Count_pre;
    xv_set(ip->product_list,
        PANEL_LIST_DELETE_ROWS, Count_pre, del_row_no, NULL);
}
.....*/
fclose(fp);
}

/*
 * 기존에 Display된 Field값들을 지운다.
 */

void del_field()
{
    if (xv_get(sub1_dwin->filename_field, PANEL_VALUE))

```

```

xv_set(subl_dwin->filename_field, PANEL_VALUE, "", NULL);

if (xv_get(subl_dwin->part_no_field, PANEL_VALUE))
xv_set(subl_dwin->part_no_field, PANEL_VALUE, "", NULL);

if (xv_get(subl_dwin->material_field, PANEL_VALUE))
xv_set(subl_dwin->material_field, PANEL_VALUE, "", NULL);

if (xv_get(subl_dwin->quantity_field, PANEL_VALUE))
xv_set(subl_dwin->quantity_field, PANEL_VALUE, "", NULL);

if (xv_get(subl_dwin->dim_field, PANEL_VALUE))
xv_set(subl_dwin->dim_field, PANEL_VALUE, "", NULL);

if (xv_get(subl_dwin->scale_field, PANEL_VALUE))
xv_set(subl_dwin->scale_field, PANEL_VALUE, "", NULL);

if (xv_get(subl_dwin->unit_field, PANEL_VALUE))
xv_set(subl_dwin->unit_field, PANEL_VALUE, "", NULL);

if (xv_get(subl_dwin->designer_field, PANEL_VALUE))
xv_set(subl_dwin->designer_field, PANEL_VALUE, "", NULL);

if (xv_get(subl_dwin->date_field, PANEL_VALUE))
xv_set(subl_dwin->date_field, PANEL_VALUE, "", NULL);

if (xv_get(subl_dwin->pload_field, PANEL_VALUE))
xv_set(subl_dwin->pload_field, PANEL_VALUE, "", NULL);

if (xv_get(subl_dwin->toolno_field, PANEL_VALUE))
xv_set(subl_dwin->toolno_field, PANEL_VALUE, "", NULL);

if (xv_get(subl_dwin->toolname_field, PANEL_VALUE))
xv_set(subl_dwin->toolname_field, PANEL_VALUE, "", NULL);

if (xv_get(subl_dwin->ncprog_field, PANEL_VALUE))
xv_set(subl_dwin->ncprog_field, PANEL_VALUE, "", NULL);

if (xv_get(subl_dwin->approval_field, PANEL_VALUE))
xv_set(subl_dwin->approval_field, PANEL_VALUE, "", NULL);
}

/*
 * Notify callback function for `product_list'.
 */
int
product_list_proc(item, string, client_data, op, event)
    Panel_item    item;
    char          *string;
    Xv_opaque     client_data;
    Panel_list_op op;
    Event         *event;
{
    subl_dwin_objects *ip = (subl_dwin_objects *) xv_get(item, XV_KEY_DATA, INSTA

    static char    product_msg[BUFSIZE];

    switch(op) {
    case PANEL_LIST_OP_DESELECT:
        break;

    case PANEL_LIST_OP_SELECT:
        sprintf(product_msg, " 제품명: %s", string);
        xv_set(ip->product_list,
            PANEL_LABEL_STRING, product_msg, NULL);
        strcpy(p_name, string);
        bzero(current_value, BUFSIZE);

```

```

        if( file_list() == 1) {
            xv_set(subl_dwin->info_msg, PANEL_LABEL_STRING, "", NULL);
            del_field();
            flist_display(1);
        }
        else
            notice_prompt(ip->product_list, NULL,
                NOTICE_FOCUS_XY,      event_x(event), event_y(event)
                NOTICE_MESSAGE_STRINGS,
                "데이터 파일이 존재하지 않습니다", NULL,
                NOTICE_BUTTON_YES,     "확인",
                NULL);

        break;

    case PANEL_LIST_OP_VALIDATE:
        break;

    case PANEL_LIST_OP_DELETE:
        break;
    }

    return XV_OK;
}

void get_current_file(count, strings)
int count;
char *strings;
{
    char    buf[BUFSIZE];
    int     bufp = 0;
    int     i = 0, j = 0;

    while((buf[j] = strings[i]) != '\0') {
        if( buf[j] == ' ' || buf[j] == '\t' ) {
            buf[j] = '\0';
            if(strchr(buf, '.') != NULL) {
                strcpy(current_value, buf);
                return;
            }
            else {
                i++;
                j = 0;
            }
        }
        else {
            i++;
            j++;
        }
    }

    if(strchr(buf, '.') != NULL)
        strcpy(current_value, buf);
}

/*
 * Database File에 Access해 Field값을 구하고 화면에 Display시킨다.
 */

void get_field()
{
    char    STREAM[BUFSIZE], Datafile[BUFSIZE];

    xv_set(subl_dwin->filename_field, PANEL_VALUE, current_value, NULL)
}

```

```

sprintf(Datafile, "%s/test.txt", directory);

if ((fp = fopen(Datafile, "r")) == NULL) {
    putchar('W07');
    xv_set(subl_dwin->info_msg,
           PANEL_LABEL_STRING,
           "데이터 파일이 존재하지 않습니다 !!!", NULL);
    return;
}

xv_set(subl_dwin->info_msg,
       PANEL_LABEL_STRING, "", NULL);

fgets(STREAM, BUFSIZE, fp);
xv_set(subl_dwin->part_no_field, PANEL_VALUE, STREAM, NULL);
fgets(STREAM, BUFSIZE, fp);
xv_set(subl_dwin->material_field, PANEL_VALUE, STREAM, NULL);
fgets(STREAM, BUFSIZE, fp);
xv_set(subl_dwin->quantity_field, PANEL_VALUE, STREAM, NULL);
fgets(STREAM, BUFSIZE, fp);
xv_set(subl_dwin->dim_field, PANEL_VALUE, STREAM, NULL);
fgets(STREAM, BUFSIZE, fp);
xv_set(subl_dwin->scale_field, PANEL_VALUE, STREAM, NULL);
fgets(STREAM, BUFSIZE, fp);
xv_set(subl_dwin->unit_field, PANEL_VALUE, STREAM, NULL);
fgets(STREAM, BUFSIZE, fp);
xv_set(subl_dwin->designer_field, PANEL_VALUE, STREAM, NULL);
fgets(STREAM, BUFSIZE, fp);
xv_set(subl_dwin->date_field, PANEL_VALUE, STREAM, NULL);
fgets(STREAM, BUFSIZE, fp);
xv_set(subl_dwin->pload_field, PANEL_VALUE, STREAM, NULL);
fgets(STREAM, BUFSIZE, fp);
xv_set(subl_dwin->toolno_field, PANEL_VALUE, STREAM, NULL);
fgets(STREAM, BUFSIZE, fp);
xv_set(subl_dwin->toolname_field, PANEL_VALUE, STREAM, NULL);
fgets(STREAM, BUFSIZE, fp);
xv_set(subl_dwin->ncprog_field, PANEL_VALUE, STREAM, NULL);
fgets(STREAM, BUFSIZE, fp);
xv_set(subl_dwin->approval_field, PANEL_VALUE, STREAM, NULL);

fclose(fp);
}

/*
 * Notify callback function for `filelist'.
 */
int
file_list_proc(item, string, client_data, op, event)
    Panel_item    item;
    char          *string;
    Xv_opaque     client_data;
    Panel_list_op op;
    Event         *event;
{
    subl_dwin_objects *ip = (subl_dwin_objects *) xv_get(item, XV_KEY_DATA, INST,

    switch(op) {
    case PANEL_LIST_OP_DESELECT:
        strcpy(current_value, "");
        break;

    case PANEL_LIST_OP_SELECT:
        get_current_file(FIELD_NUM, string);
        del_field();
        get_field();
        break;
    case PANEL_LIST_OP_VALIDATE:

```



```

                break:

        case PANEL_LIST_OP_DELETE:
                break:
        }
        return XV_OK;
}

/*
 * Notify callback function for `del_button'.
 */
void
del_proc(item, event)
        Panel_item    item;
        Event          *event;
{
        subl_dwin_objects    *ip = (subl_dwin_objects *) xv_get(item, XV_KEY_DATA, INST

        static char    buff[BUFSIZE];
        int    result, Count, i, j;

        if( !current_value[0]) {
                notice_prompt(ip->subl_control, NULL,
                NOTICE_FOCUS_XY,    event_x(event), event_y(event),
                NOTICE_MESSAGE_STRINGS, "선택된 파일이 없습니다.",
                "파일을 먼저 선택하십시오", NULL,
                NOTICE_BUTTON_YES,    "확인",
                NULL);
        }
        else {
                result = notice_prompt(ip->subl_control, NULL,
                NOTICE_FOCUS_XY,    event_x(event), event_y(event),
                NOTICE_MESSAGE_STRINGS, "해당 File을 지우려 합니다.",
                "다시 한번 더 확인하십시오", NULL,
                NOTICE_BUTTON_YES,    "삭제 확인",
                NOTICE_BUTTON_NO,    "삭제 취소",
                NULL);

                if (result == NOTICE_YES) {
                        sprintf(buff, "/usr/bin/rm %s/%s" ,directory, current_value);

                        if( system(buff) == -1 ) {
                                perror("execl");
                                exit(0);
                        }

                        system("");
                        file_list();
                        flist_display(1);
                }
        }
}

/*
 * Notify callback function for `pdel_button'.
 */
void
pdel_proc(item, event)
        Panel_item    item;
        Event          *event;
{
        subl_dwin_objects    *ip = (subl_dwin_objects *) xv_get(item, XV_KEY_DATA, INSTA

        int    result, cnt = 0;
        int    Count, i;

```

```

static char    buff[BUFSIZE];

if( !p_name[0] ) {
    notice_prompt(ip->subl_control, NULL,
        NOTICE_FOCUS_XY,    event_x(event), event_y(event),
        NOTICE_MESSAGE_STRINGS, "선택된 제품이 없습니다",
        "제품을 먼저 선택하십시오", NULL,
        NOTICE_BUTTON_YES,    "확인",
        NULL);
}
else {
    result = notice_prompt(ip->subl_control, NULL,
        NOTICE_FOCUS_XY,    event_x(event), event_y(event),
        NOTICE_MESSAGE_STRINGS, "해당 제품내의 파일을 모두 지우려 합니다",
        "    다시 한번 더 확인하십시오", NULL,
        NOTICE_BUTTON_YES,    "삭제 확인",
        NOTICE_BUTTON_NO,    "삭제 취소",
        NULL);

    if (result == NOTICE_YES) {
        sprintf(buff, "/usr/bin/rm -r %s%s", ROOTDIR, p_name);

        if( system(buff) == -1 ) {
            perror("execl");
            exit(0);
        }

        system("");

        del_plist();
        search_file(1);
        del_flist();
    }
}

}

/*
 * Notify callback function for `mv_button'.
 */
void
mv_proc(item, event)
    Panel_item    item;
    Event          *event;
{
    subl_dwin_objects    *ip = (subl_dwin_objects *) xv_get(item, XV_KEY_DATA, INSTA

if( !current_value[0] ) {
    notice_prompt(ip->subl_control, NULL,
        NOTICE_FOCUS_XY,    event_x(event), event_y(event),
        NOTICE_MESSAGE_STRINGS, "선택된 파일이 없습니다",
        "이름을 변경할 파일을 먼저 선택하십시오", NULL,
        NOTICE_BUTTON_YES,    "확인",
        NULL);
}
else {
    xv_set(pop_popup->popup,
        XV_X,    715,
        XV_Y,    170,
        XV_SHOW, TRUE, NULL);
    xv_set(pop_popup->filename_msg,
        PANEL_LABEL_STRING, current_value,
        NULL);
}
}

```

```

}

/*
 * Notify callback function for `info_button'.
 */
void
pdpm_prc(item, event)
    Panel_item    item;
    Event         *event;
{
    subl_dwin_objects *ip = (subl_dwin_objects *) xv_get(item, XV_KEY_DATA, INST

    char    buff[BUFSIZE];

    if( !current_value[0] ) {
        notice_prompt(ip->subl_control, NULL,
            NOTICE_FOCUS_XY,    event_x(event), event_y(event),
            NOTICE_MESSAGE_STRINGS, "선택된 파일이 없습니다",
            NOTICE_BUTTON_YES,   "조회할 파일을 먼저 선택하십시오", NULL,
            NULL);
    }
    else {
        sprintf(buff, "cp %s/%s .", directory, current_value);

        if( system(buff) == -1 ) {
            perror("execl");
            exit(0);
        }

        system("");

        sprintf(buff, "pm -c graphics_m %s -Wp 330 240",
            current_value);

        if( system(buff) == -1 ) {
            perror("execl");
            exit(0);
        }

        system("");

        sprintf(buff, "rm %s", current_value);

        if( system(buff) == -1 ) {
            perror("execl");
            exit(0);
        }

        system("");
    }
}

```

```

/*
 * Notify callback function for `print_button'.
 */
void
print_proc(item, event)
    Panel_item    item;
    Event         *event;
{
    subl_dwin_objects *ip = (subl_dwin_objects *) xv_get(item, XV_KEY_DATA, INSTA

    if(1) !

```

```
notice_prompt(ip->subl_control, NULL,  
NOTICE_FOCUS_XY, event_x(event), event_y(event),  
NOTICE_MESSAGE_STRINGS, "Error 발생 !",  
"Plotter가 연결되지 않음", NULL,  
NOTICE_BUTTON_YES, "확인",  
NULL);
```

```
}
```

```
}
```

```

#include <stdio.h>
#include <sys/param.h>
#include <sys/types.h>
#include <xview/xview.h>
#include <xview/canvas.h>
#include <xview/icon_load.h>
#include <xview/panel.h>
#include <xview/scrollbar.h>
#include <xview/svimage.h>
#include <xview/termsw.h>
#include <xview/text.h>
#include <xview/tty.h>
#include <xview/xv_xrect.h>
#include <gcm.h>
#include "subl_ui.h"

/*
 * Initialize an instance of object `dwin'.
 */
subl_dwin_objects *
subl_dwin_objects_initialize(ip, owner)
    subl_dwin_objects *ip;
    Xv_opaque owner;
{
    if (!ip && !(ip = (subl_dwin_objects *) calloc(1, sizeof (subl_dwin_objects))))
        return (subl_dwin_objects *) NULL;
    if (!ip->dwin)
        ip->dwin = subl_dwin_dwin_create(ip, owner);
    if (!ip->pop_control)
        ip->pop_control = subl_dwin_pop_control_create(ip, ip->dwin);
    if (!ip->pop_control_msg)
        ip->pop_control_msg = subl_dwin_pop_control_msg_create(ip, ip->pop_control);
    if (!ip->subl_control)
        ip->subl_control = subl_dwin_subl_control_create(ip, ip->dwin);
    if (!ip->flist_set)
        ip->flist_set = subl_dwin_flist_set_create(ip, ip->subl_control);
    if (!ip->info_button)
        ip->info_button = subl_dwin_info_button_create(ip, ip->subl_control);
    if (!ip->del_button)
        ip->del_button = subl_dwin_del_button_create(ip, ip->subl_control);
    if (!ip->pdel_button)
        ip->pdel_button = subl_dwin_pdel_button_create(ip, ip->subl_control);
    if (!ip->mv_button)
        ip->mv_button = subl_dwin_mv_button_create(ip, ip->subl_control);
    if (!ip->print_button)
        ip->print_button = subl_dwin_print_button_create(ip, ip->subl_control);
    if (!ip->drw_nc_msg)
        ip->drw_nc_msg = subl_dwin_drw_nc_msg_create(ip, ip->subl_control);
    if (!ip->subl_control_msg)
        ip->subl_control_msg = subl_dwin_subl_control_msg_create(ip, ip->subl_control);
    if (!ip->list_control1)
        ip->list_control1 = subl_dwin_list_control1_create(ip, ip->dwin);
    if (!ip->product_list)
        ip->product_list = subl_dwin_product_list_create(ip, ip->list_control1);
    if (!ip->list_control2)
        ip->list_control2 = subl_dwin_list_control2_create(ip, ip->dwin);
    if (!ip->filelist)
        ip->filelist = subl_dwin_filelist_create(ip, ip->list_control2);
    if (!ip->context_control)
        ip->context_control = subl_dwin_context_control_create(ip, ip->dwin);
    if (!ip->context_msg)
        ip->context_msg = subl_dwin_context_msg_create(ip, ip->context_control);
    if (!ip->filename_field)
        ip->filename_field = subl_dwin_filename_field_create(ip, ip->context_control);
    if (!ip->info_msg)
        ip->info_msg = subl_dwin_info_msg_create(ip, ip->context_control);
    if (!ip->part_no_field)
        ip->part_no_field = subl_dwin_part_no_field_create(ip, ip->context_control);
}

```



```

    if (!ip->date_field)
        ip->date_field = subl_dwin_date_field_create(ip, ip->context_control);
    if (!ip->material_field)
        ip->material_field = subl_dwin_material_field_create(ip, ip->context_control);
    if (!ip->pload_field)
        ip->pload_field = subl_dwin_pload_field_create(ip, ip->context_control);
    if (!ip->quantity_field)
        ip->quantity_field = subl_dwin_quantity_field_create(ip, ip->context_control);
    if (!ip->toolno_field)
        ip->toolno_field = subl_dwin_toolno_field_create(ip, ip->context_control);
    if (!ip->dim_field)
        ip->dim_field = subl_dwin_dim_field_create(ip, ip->context_control);
    if (!ip->toolname_field)
        ip->toolname_field = subl_dwin_toolname_field_create(ip, ip->context_control);
    if (!ip->scale_field)
        ip->scale_field = subl_dwin_scale_field_create(ip, ip->context_control);
    if (!ip->ncprog_field)
        ip->ncprog_field = subl_dwin_ncprog_field_create(ip, ip->context_control);
    if (!ip->approval_field)
        ip->approval_field = subl_dwin_approval_field_create(ip, ip->context_control);
    if (!ip->unit_field)
        ip->unit_field = subl_dwin_unit_field_create(ip, ip->context_control);
    if (!ip->designer_field)
        ip->designer_field = subl_dwin_designer_field_create(ip, ip->context_control);
    return ip;
}

/*
 * Create object `dwin' in the specified instance.
 */
Xv_opaque
subl_dwin_dwin_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, FRAME_CMD,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_WIDTH, 1140,
                    XV_HEIGHT, 753,
                    XV_SHOW, FALSE,
                    FRAME_SHOW_FOOTER, FALSE,
                    FRAME_SHOW_RESIZE_CORNER, FALSE,
                    FRAME_CMD_PUSHPIN_IN, TRUE,
                    NULL);
    gcm_initialize_colors(obj, "Black", "Yellow");
    xv_set(xv_get(obj, FRAME_CMD_PANEL), WIN_SHOW, FALSE, NULL);
    return obj;
}

/*
 * Create object `pop_control' in the specified instance.
 */
Xv_opaque
subl_dwin_pop_control_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 0,
                    XV_Y, 0,
                    XV_WIDTH, 150,

```

```

        XV_HEIGHT, 100,
        WIN_BORDER, TRUE,
        NULL);
gcm_initialize_colors(obj, "Dark Slate Gray", "Yellow");
return obj;
}

/*
 * Create object `pop_control_msg' in the specified instance.
 */
Xv_opaque
subl_dwin_pop_control_msg_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;
    Xv_opaque    pop_control_msg_image;
    static unsigned short pop_control_msg_bits[] = {
#include "file_mng.icon"
    };

    pop_control_msg_image = xv_create(XV_NULL, SERVER_IMAGE,
        SERVER_IMAGE_BITS, pop_control_msg_bits,
        SERVER_IMAGE_DEPTH, 1,
        XV_WIDTH, 64,
        XV_HEIGHT, 64,
        NULL);
    obj = xv_create(owner, PANEL_MESSAGE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 40,
        XV_Y, 16,
        XV_WIDTH, 64,
        XV_HEIGHT, 64,
        PANEL_ITEM_COLOR, gcm_color_index("Dark Turquoise"),
        PANEL_LABEL_IMAGE, pop_control_msg_image,
        PANEL_LABEL_BOLD, FALSE,
        NULL);
    return obj;
}

/*
 * Create object `subl_control' in the specified instance.
 */
Xv_opaque
subl_dwin_subl_control_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 150,
        XV_Y, 0,
        XV_WIDTH, WIN_EXTEND_TO_EDGE,
        XV_HEIGHT, 100,
        WIN_BORDER, TRUE,
        NULL);
    gcm_initialize_colors(obj, "Dark Slate Gray", "White");
    return obj;
}

/*
 * Create object `flist_set' in the specified instance.
 */

```

```

Xv_opaque
subl_dwin_flist_set_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern int      flist_set_proc();
    Xv_opaque      obj;

    obj = xv_create(owner, PANEL_CHOICE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 24,
        XV_Y, 24,
        XV_WIDTH, 268,
        XV_HEIGHT, 26,
        PANEL_VALUE_X, 103,
        PANEL_VALUE_Y, 24,
        PANEL_CHOICE_NROWS, 1,
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_ITEM_COLOR, gcm_color_index("Yellow"),
        PANEL_LABEL_STRING, "조회 파일:",
        PANEL_NOTIFY_PROC, flist_set_proc,
        PANEL_CHOICE_STRINGS,
            "도면 파일",
            "NC Data 파일",
            0,
        PANEL_CHOICE_COLOR, 0, gcm_color_index("Blue"),
        PANEL_CHOICE_COLOR, 1, gcm_color_index("Blue")
        NULL);
    return obj;
}

/*
 * Create object `info_button' in the specified instance.
 */
Xv_opaque
subl_dwin_info_button_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern void    pdpm_proc();
    Xv_opaque      obj;

    obj = xv_create(owner, PANEL_BUTTON,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 344,
        XV_Y, 24,
        XV_WIDTH, 113,
        XV_HEIGHT, 21,
        PANEL_ITEM_COLOR, gcm_color_index("Blue"),
        PANEL_LABEL_STRING, "설계도면 확인",
        PANEL_NOTIFY_PROC, pdpm_proc,
        NULL);
    return obj;
}

/*
 * Create object `del_button' in the specified instance.
 */
Xv_opaque
subl_dwin_del_button_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern void    del_proc();
    Xv_opaque      obj;

```

```

        obj = xv_create(owner, PANEL_BUTTON,
            XV_KEY_DATA, INSTANCE, ip,
            XV_X, 500,
            XV_Y, 24,
            XV_WIDTH, 85,
            XV_HEIGHT, 21,
            PANEL_ITEM_COLOR, gcm_color_index("Blue"),
            PANEL_LABEL_STRING, "File 삭제",
            PANEL_NOTIFY_PROC, del_proc,
            NULL);
    return obj;
}

/*
 * Create object `pdel_button' in the specified instance.
 */
Xv_opaque
subl_dwin_pdel_button_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern void      pdel_proc();
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_BUTTON,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 608,
        XV_Y, 24,
        XV_WIDTH, 108,
        XV_HEIGHT, 21,
        PANEL_ITEM_COLOR, gcm_color_index("Red"),
        PANEL_LABEL_STRING, "Product 삭제",
        PANEL_NOTIFY_PROC, pdel_proc,
        NULL);
    return obj;
}

/*
 * Create object `mv_button' in the specified instance.
 */
Xv_opaque
subl_dwin_mv_button_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern void      mv_proc();
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_BUTTON,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 744,
        XV_Y, 24,
        XV_WIDTH, 99,
        XV_HEIGHT, 21,
        PANEL_ITEM_COLOR, gcm_color_index("Blue"),
        PANEL_LABEL_STRING, "File Rename",
        PANEL_NOTIFY_PROC, mv_proc,
        NULL);
    return obj;
}

/*
 * Create object `print_button' in the specified instance
 */
Xv_opaque

```

```

subl_dwin_print_button_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern void      print_proc();
    Xv_opaque      obj;

    obj = xv_create(owner, PANEL_BUTTON,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 868,
        XV_Y, 24,
        XV_WIDTH, 78,
        XV_HEIGHT, 21,
        PANEL_ITEM_COLOR, gcm_color_index("Blue"),
        PANEL_LABEL_STRING, "도면출도",
        PANEL_NOTIFY_PROC, print_proc,
        NULL);
    return obj;
}

/*
 * Create object `drw_nc_msg' in the specified instance.
 */
Xv_opaque
subl_dwin_drw_nc_msg_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_MESSAGE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 24,
        XV_Y, 68,
        XV_WIDTH, 218,
        XV_HEIGHT, 16,
        PANEL_ITEM_COLOR, gcm_color_index("Yellow"),
        PANEL_LABEL_STRING, "도면 파일이 선택되고 관리됩니다",
        PANEL_LABEL_BOLD, TRUE,
        NULL);
    return obj;
}

/*
 * Create object `subl_control_msg' in the specified instance.
 */
Xv_opaque
subl_dwin_subl_control_msg_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_MESSAGE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 564,
        XV_Y, 72,
        XV_WIDTH, 295,
        XV_HEIGHT, 16,
        PANEL_ITEM_COLOR, gcm_color_index("Yellow"),
        PANEL_LABEL_STRING, "File을 조회한 후 원하는 Button을 누르시오",
        PANEL_LABEL_BOLD, TRUE,
        NULL);
    return obj;
}

```



```

/*
 * Create object `list_controll' in the specified instance.
 */
Xv_opaque
subl_dwin_list_controll_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 0,
                    XV_Y, 100,
                    XV_WIDTH, 150,
                    XV_HEIGHT, WIN_EXTEND_TO_EDGE,
                    WIN_BORDER, TRUE,
                    NULL);
    gcm_initialize_colors(obj, "Dark Slate Gray", "White");
    return obj;
}

/*
 * Create object `product_list' in the specified instance.
 */
Xv_opaque
subl_dwin_product_list_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern int    product_list_proc();
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_LIST,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 0,
                    XV_Y, 4,
                    PANEL_LIST_WIDTH, 130,
                    XV_HEIGHT, 640,
                    PANEL_LIST_DISPLAY_ROWS, 29,
                    PANEL_ITEM_COLOR, gcm_color_index("Medium Spring Green"),
                    PANEL_LABEL_STRING, " 제품명 :",
                    PANEL_LAYOUT, PANEL_VERTICAL,
                    PANEL_READ_ONLY, FALSE,
                    PANEL_CHOOSE_ONE, TRUE,
                    PANEL_CHOOSE_NONE, TRUE,
                    PANEL_NOTIFY_PROC, product_list_proc,
                    NULL);
    return obj;
}

/*
 * Create object `list_control2' in the specified instance.
 */
Xv_opaque
subl_dwin_list_control2_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 150,
                    XV_Y, 100,

```

```

        XV_WIDTH, 450,
        XV_HEIGHT, WIN_EXTEND_TO_EDGE,
        WIN_BORDER, TRUE,
        NULL);
    gcm_initialize_colors(obj, "Dark Slate Gray", "Yellow")
    return obj;
}

/*
 * Create object `filelist' in the specified instance.
 */
Xv_opaque
subl_dwin_filelist_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern int      file_list_proc();
    Xv_opaque      obj;

    obj = xv_create(owner, PANEL_LIST,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 4,
        XV_Y, 4,
        PANEL_LIST_WIDTH, 425,
        XV_HEIGHT, 649,
        PANEL_LIST_DISPLAY_ROWS, 29,
        PANEL_ITEM_COLOR, gcm_color_index("White"),
        PANEL_LAYOUT, PANEL_VERTICAL,
        PANEL_READ_ONLY, TRUE,
        PANEL_CHOOSE_ONE, TRUE,
        PANEL_CHOOSE_NONE, TRUE,
        PANEL_NOTIFY_PROC, file_list_proc,
        NULL);
    return obj;
}

/*
 * Create object `context_control' in the specified instance.
 */
Xv_opaque
subl_dwin_context_control_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque      obj;

    obj = xv_create(owner, PANEL,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 600,
        XV_Y, 100,
        XV_WIDTH, WIN_EXTEND_TO_EDGE,
        XV_HEIGHT, WIN_EXTEND_TO_EDGE,
        WIN_BORDER, TRUE,
        NULL);
    gcm_initialize_colors(obj, "Black", NULL);
    return obj;
}

/*
 * Create object `context_msg' in the specified instance.
 */
Xv_opaque
subl_dwin_context_msg_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;

```

```

{
    Xv_opaque      obj;

    obj = xv_create(owner, PANEL_MESSAGE,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 176,
                    XV_Y, 36,
                    XV_WIDTH, 120,
                    XV_HEIGHT, 16,
                    PANEL_ITEM_COLOR, gcm_color_index("Yellow"),
                    PANEL_LABEL_STRING, "<< 설계 정보 >>",
                    PANEL_LABEL_BOLD, TRUE,
                    NULL);
    return obj;
}

/*
 * Create object `filename_field' in the specified instance
 */
Xv_opaque
subl_dwin_filename_field_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_TEXT,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 144,
                    XV_Y, 76,
                    XV_WIDTH, 205,
                    XV_HEIGHT, 18,
                    PANEL_VALUE_X, 209,
                    PANEL_VALUE_Y, 76,
                    PANEL_VALUE_DISPLAY_LENGTH, 20,
                    PANEL_VALUE_STORED_LENGTH, 20,
                    PANEL_ITEM_COLOR, gcm_color_index("Green"),
                    PANEL_LABEL_STRING, "파일명 :",
                    PANEL_LAYOUT, PANEL_HORIZONTAL,
                    PANEL_READ_ONLY, FALSE,
                    NULL);
    return obj;
}

/*
 * Create object `info_msg' in the specified instance.
 */
Xv_opaque
subl_dwin_info_msg_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_MESSAGE,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 104,
                    XV_Y, 120,
                    XV_WIDTH, 1,
                    XV_HEIGHT, 16,
                    PANEL_ITEM_COLOR, gcm_color_index("Red"),
                    PANEL_LABEL_BOLD, TRUE,
                    NULL);
    return obj;
}

```

```

/*
 * Create object `part_no_field' in the specified instance.
 */
Xv_opaque
subl_dwin_part_no_field_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_TEXT,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 60,
                    XV_Y, 184,
                    XV_WIDTH, 121,
                    XV_HEIGHT, 18,
                    PANEL_VALUE_X, 139,
                    PANEL_VALUE_Y, 184,
                    PANEL_VALUE_DISPLAY_LENGTH, 5,
                    PANEL_VALUE_STORED_LENGTH, 5,
                    PANEL_ITEM_COLOR, gcm_color_index("green"),
                    PANEL_LABEL_STRING, "부품번호 :",
                    PANEL_LAYOUT, PANEL_HORIZONTAL,
                    PANEL_READ_ONLY, FALSE,
                    NULL);
    return obj;
}

/*
 * Create object `date_field' in the specified instance.
 */
Xv_opaque
subl_dwin_date_field_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_TEXT,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 260,
                    XV_Y, 184,
                    XV_WIDTH, 149,
                    XV_HEIGHT, 18,
                    PANEL_VALUE_X, 339,
                    PANEL_VALUE_Y, 184,
                    PANEL_VALUE_DISPLAY_LENGTH, 10,
                    PANEL_VALUE_STORED_LENGTH, 10,
                    PANEL_ITEM_COLOR, gcm_color_index("green"),
                    PANEL_LABEL_STRING, "설계날짜 :",
                    PANEL_LAYOUT, PANEL_HORIZONTAL,
                    PANEL_READ_ONLY, FALSE,
                    NULL);
    return obj;
}

/*
 * Create object `material_field' in the specified instance
 */
Xv_opaque
subl_dwin_material_field_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

```

```

    obj = xv_create(owner, PANEL_TEXT,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 56,
        XV_Y, 264,
        XV_WIDTH, 135,
        XV_HEIGHT, 18,
        PANEL_VALUE_X, 135,
        PANEL_VALUE_Y, 264,
        PANEL_VALUE_DISPLAY_LENGTH, 8,
        PANEL_VALUE_STORED_LENGTH, 8,
        PANEL_ITEM_COLOR, gcm_color_index("Green"),
        PANEL_LABEL_STRING, "부품재질 :",
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_READ_ONLY, FALSE,
        NULL);
    return obj;
}

/*
 * Create object `pload_field' in the specified instance.
 */
Xv_opaque
subl_dwin_pload_field_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_TEXT,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 256,
        XV_Y, 264,
        XV_WIDTH, 121,
        XV_HEIGHT, 18,
        PANEL_VALUE_X, 335,
        PANEL_VALUE_Y, 264,
        PANEL_VALUE_DISPLAY_LENGTH, 5,
        PANEL_VALUE_STORED_LENGTH, 5,
        PANEL_ITEM_COLOR, gcm_color_index("Green"),
        PANEL_LABEL_STRING, "공정순서 :",
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_READ_ONLY, FALSE,
        NULL);
    return obj;
}

/*
 * Create object `quantity_field' in the specified instance
 */
Xv_opaque
subl_dwin_quantity_field_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_TEXT,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 56,
        XV_Y, 316,
        XV_WIDTH, 121,
        XV_HEIGHT, 18,
        PANEL_VALUE_X, 135,
        PANEL_VALUE_Y, 316,
        PANEL_VALUE_DISPLAY_LENGTH, 3,

```



```

        PANEL_VALUE_STORED_LENGTH, 3,
        PANEL_ITEM_COLOR, gcm_color_index("green")
        PANEL_LABEL_STRING, "부품수량 :",
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_READ_ONLY, FALSE,
        NULL);
    return obj;
}

/*
 * Create object `toolno_field' in the specified instance.
 */
Xv_opaque
subl_dwin_toolno_field_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_TEXT,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 256,
        XV_Y, 316,
        XV_WIDTH, 149,
        XV_HEIGHT, 18,
        PANEL_VALUE_X, 335,
        PANEL_VALUE_Y, 316,
        PANEL_VALUE_DISPLAY_LENGTH, 10,
        PANEL_VALUE_STORED_LENGTH, 10,
        PANEL_ITEM_COLOR, gcm_color_index("green")
        PANEL_LABEL_STRING, "공구번호 :",
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_READ_ONLY, FALSE,
        NULL);
    return obj;
}

/*
 * Create object `dim_field' in the specified instance.
 */
Xv_opaque
subl_dwin_dim_field_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_TEXT,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 56,
        XV_Y, 372,
        XV_WIDTH, 149,
        XV_HEIGHT, 18,
        PANEL_VALUE_X, 135,
        PANEL_VALUE_Y, 372,
        PANEL_VALUE_DISPLAY_LENGTH, 10,
        PANEL_VALUE_STORED_LENGTH, 10,
        PANEL_ITEM_COLOR, gcm_color_index("Green")
        PANEL_LABEL_STRING, "완성횟수 :",
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_READ_ONLY, FALSE,
        NULL);
    return obj;
}

/*

```

```

* Create object `toolname_field' in the specified instance

*/
Xv_opaque
subl_dwin_toolname_field_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_TEXT,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 256,
        XV_Y, 372,
        XV_WIDTH, 219,
        XV_HEIGHT, 18,
        PANEL_VALUE_X, 335,
        PANEL_VALUE_Y, 372,
        PANEL_VALUE_DISPLAY_LENGTH, 20,
        PANEL_VALUE_STORED_LENGTH, 20,
        PANEL_ITEM_COLOR, gcm_color_index("green"),
        PANEL_LABEL_STRING, "공구명 :",
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_READ_ONLY, FALSE,
        NULL);
    return obj;
}

/*
* Create object `scale_field' in the specified instance.

*/
Xv_opaque
subl_dwin_scale_field_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_TEXT,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 56,
        XV_Y, 428,
        XV_WIDTH, 135,
        XV_HEIGHT, 18,
        PANEL_VALUE_X, 135,
        PANEL_VALUE_Y, 428,
        PANEL_VALUE_DISPLAY_LENGTH, 8,
        PANEL_VALUE_STORED_LENGTH, 8,
        PANEL_ITEM_COLOR, gcm_color_index("green"),
        PANEL_LABEL_STRING, "도면척도 :",
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_READ_ONLY, FALSE,
        NULL);
    return obj;
}

/*
* Create object `ncprog_field' in the specified instance.

*/
Xv_opaque
subl_dwin_ncprog_field_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

```

```

    obj = xv_create(owner, PANEL_TEXT,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 256,
        XV_Y, 428,
        XV_WIDTH, 177,
        XV_HEIGHT, 18,
        PANEL_VALUE_X, 363,
        PANEL_VALUE_Y, 428,
        PANEL_VALUE_DISPLAY_LENGTH, 10,
        PANEL_VALUE_STORED_LENGTH, 10,
        PANEL_ITEM_COLOR, gcm_color_index("Green"),
        PANEL_LABEL_STRING, "NC프로그래밍 :",
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_READ_ONLY, FALSE,
        NULL);
    return obj;
}

/*
 * Create object `approval_field' in the specified instance
 */
Xv_opaque
subl_dwin_approval_field_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_TEXT,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 256,
        XV_Y, 488,
        XV_WIDTH, 191,
        XV_HEIGHT, 18,
        PANEL_VALUE_X, 307,
        PANEL_VALUE_Y, 488,
        PANEL_VALUE_DISPLAY_LENGTH, 20,
        PANEL_VALUE_STORED_LENGTH, 20,
        PANEL_ITEM_COLOR, gcm_color_index("green"),
        PANEL_LABEL_STRING, "검도 :",
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_READ_ONLY, FALSE,
        NULL);
    return obj;
}

/*
 * Create object `unit_field' in the specified instance.
 */
Xv_opaque
subl_dwin_unit_field_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_TEXT,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 56,
        XV_Y, 492,
        XV_WIDTH, 121,
        XV_HEIGHT, 18,
        PANEL_VALUE_X, 135,
        PANEL_VALUE_Y, 492,
        PANEL_VALUE_DISPLAY_LENGTH, 4,
        PANEL_VALUE_STORED_LENGTH, 4,

```

```

        PANEL_ITEM_COLOR, gcm_color_index("Green"),
        PANEL_LABEL_STRING, "단위      :",
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_READ_ONLY, FALSE,
        NULL);
    return obj;
}

/*
 * Create object `designer_field' in the specified instance
 */
Xv_opaque
subl_dwin_designer_field_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_TEXT,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 168,
        XV_Y, 576,
        XV_WIDTH, 205,
        XV_HEIGHT, 18,
        PANEL_VALUE_X, 233,
        PANEL_VALUE_Y, 576,
        PANEL_VALUE_DISPLAY_LENGTH, 20,
        PANEL_VALUE_STORED_LENGTH, 20,
        PANEL_ITEM_COLOR, gcm_color_index("Green"),
        PANEL_LABEL_STRING, "설계자      :",
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_READ_ONLY, FALSE,
        NULL);
    return obj;
}

```

```

#include <stdio.h>
#include <unistd.h>
#include <strings.h>
#include <errno.h>
#include <sys/param.h>
#include <sys/stat.h>
#include <sys/dir.h>
#include <sys/types.h>
#include <sys/fcntl.h>
#include <sys/termios.h>
#include <xview/xview.h>
#include <xview/panel.h>
#include <xview/textsw.h>
#include <xview/xv_xrect.h>
#include <xview/notice.h>
#include <xview/svrmage.h>
#include <gdd.h>
#include <xview/notify.h>

#include "main.h"
#include "sub2_ui.h"

sub2_dncwin_objects      *sub2_dncwin;

extern int      file_list();
extern void     flist_display();
extern void     get_current_file();

char  p_name[BUFSIZE];      /* Product Name          */
char  det_cadm[BUFSIZE];    /* DRW DIR or NC DIR     */
char  file_name[BUFSIZE];   /* *.drw 이나 *.dat나    */
char  current_value[BUFSIZE]; /* 현재 선택된 Filename  */
int   ncpu;                 /* NC로 전송이냐 PC로 전송이냐 */

struct Send_data {
    char  machine_value[10]; /* 현재 선택된 가공기계 */
    char  baudrate[10];
    char  databit[10];
    char  stopbit[10];
    char  parity[10];
} send_data;

#ifdef MAIN

/*
 * Instance XV_KEY_DATA key.  An instance is a set of related
 * user interface objects.  A pointer to an object's instance
 * is stored under this key in every object.  This must be a
 * global variable.
 */
Attr_attribute  INSTANCE;

void
main(argc, argv)
    int      argc;
    char     **argv;
{
    sub2_dncwin_objects      *sub2_dncwin;

    /*
     * Initialize XView.
     */
    xv_init(XV_INIT_ARGC_PTR_ARGV, &argc, argv,
            XV_USE_LOCALE, TRUE,
            0);
    INSTANCE = xv_unique_key();

    /*

```



```

    * Initialize user interface components.
    */
    sub2_dncwin = sub2_dncwin_objects_initialize(NULL, NULL);

    /*
    * Turn control over to XView.
    */
    xv_main_loop(sub2_dncwin->dncwin);
    exit(0);
}

#endif

/*
* Notify callback function for `product_list'.
*/
int
dnc_plist_proc(item, string, client_data, op, event)
    Panel_item    item;
    char          *string;
    Xv_opaque     client_data;
    Panel_list_op op;
    Event         *event;
{
    sub2_dncwin_objects    *ip = (sub2_dncwin_objects *) xv_get(item, XV_KEY_DATA, IN
);

    static char    product_msg[BUFSIZE];

    switch(op) {
    case PANEL_LIST_OP_DESELECT:
        break;

    case PANEL_LIST_OP_SELECT:
        sprintf(product_msg, " 제품명: %s", string);
        xv_set(ip->product_list,
            PANEL_LABEL_STRING, product_msg, NULL);
        if(xv_get(ip->file_editor, TEXTSW_FILE) != NULL)
            xv_set(ip->file_editor, TEXTSW_FILE, NULLFILE, NULL);
        strcpy(det_cadm, NCDIR);
        strcpy(file_name, NCFILE);
        strcpy(p_name, string);
        bzero(current_value, BUFSIZE);

        if( file_list() == 1)
            flist_display(2);
        else
            notice_prompt(ip->product_list, NULL,
                NOTICE_FOCUS_XY,    event_x(event), event_y(event),
                NOTICE_MESSAGE_STRINGS,
                "데이터 파일이 존재하지 않습니다", NULL,
                NOTICE_BUTTON_YES,    "확인",
                NULL);

        break;
    case PANEL_LIST_OP_VALIDATE:
        break;

    case PANEL_LIST_OP_DELETE:
        break;
    }
    return XV_OK;
}

/*
* Notify callback function for `filelist'.
*/
int
dnc_flist_proc(item, string, client_data, op, event)

```

```

    Panel_item    item;
    char          *string;
    Xv_opaque     client_data;
    Panel_list_op op;
    Event        *event;
{
    sub2_dncwin_objects    *ip = (sub2_dncwin_objects *) xv_get(item, XV_KEY_DATA, IN
);

    char    directory[BUFSIZE];    /* 선택된 Directory */
    char    buff[BUFSIZE];

    switch(op) {
    case PANEL_LIST_OP_DESELECT:
        break;

    case PANEL_LIST_OP_SELECT:
        get_current_file(FIELD_NUM, string);
        sprintf(directory, "%s%s%s", ROOTDIR, p_name, det_cadm);
        sprintf(buff, "%s/%s", directory, current_value);
        xv_set(ip->file_editor, TEXTSW_FILE, buff, NULL);
        break;

    case PANEL_LIST_OP_VALIDATE:
        break;

    case PANEL_LIST_OP_DELETE:
        break;
    }
    return XV_OK;
}

/*
 * Notify callback function for `send_type1'.
 */
void
sendtype_proc(item, value, event)
    Panel_item    item;
    int          value;
    Event        *event;
{
    sub2_dncwin_objects    *ip = (sub2_dncwin_objects *) xv_get(item, XV_KEY_DATA, IN
);

    Xv_opaque     msg_image;
    static unsigned short    msg_bits0[] = {
#include "machine_msg.icon"
    };
    static unsigned short    msg_bits1[] = {
#include "pc_msg.icon"
    };
    static unsigned short    msg_bits2[] = {
#include "tape_msg.icon"
    };

    switch(value) {
    case 0 :
        ncpc = 0;
        msg_image = xv_create(XV_NULL, SERVER_IMAGE,
            SERVER_IMAGE_BITS, msg_bits0,
            SERVER_IMAGE_DEPTH, 1,
            XV_WIDTH, 64,
            XV_HEIGHT, 64,
            NULL);
        break;
    case 1 :
        ncpc = 1;

```

```

        msg_image = xv_create(XV_NULL, SERVER_IMAGE,
            SERVER_IMAGE_BITS, msg_bits1,
            SERVER_IMAGE_DEPTH, 1,
            XV_WIDTH, 64,
            XV_HEIGHT, 64,
            NULL);
        break;
    case 2 :
        ncpu = 2;
        msg_image = xv_create(XV_NULL, SERVER_IMAGE,
            SERVER_IMAGE_BITS, msg_bits2,
            SERVER_IMAGE_DEPTH, 1,
            XV_WIDTH, 64,
            XV_HEIGHT, 64,
            NULL);
        notice_prompt(ip->sub1_control, NULL,
            NOTICE_FOCUS_XY, event_x(event), event_y(event),
            NOTICE_MESSAGE_STRINGS, " Error 발생 !",
            "천공 테이프가 연결되지 않음", NULL,
            NOTICE_BUTTON_YES, "확인",
            NULL);
        break;
    }
    xv_set(ip->send_condition_msg, PANEL_LABEL_IMAGE, msg_image, NULL);
}

/*
 * Notify callback function for `baud_slider'.
 */
void
baudrate_proc(item, value, event)
    Panel_item item;
    int value;
    Event *event;
{
    sub2_dncwin_objects *ip = (sub2_dncwin_objects *) xv_get(item, XV_KEY_DATA, IN
);

    switch(value) {
    case 0 :
        strcpy(send_data.baudrate, "B9600");
        break;
    case 1 :
        strcpy(send_data.baudrate, "B300");
        break;
    case 2 :
        strcpy(send_data.baudrate, "B1200");
        break;
    case 3 :
        strcpy(send_data.baudrate, "B2400");
        break;
    case 4 :
        strcpy(send_data.baudrate, "B4800");
        break;
    case 5 :
        strcpy(send_data.baudrate, "B19200");
        break;
    }
    fprintf(stderr, "baudrate: %s\n", send_data.baudrate);
}

/*
 * Notify callback function for `databit_set'.
 */
void

```

```

databit_proc(item, value, event)
    Panel_item    item;
    int           value;
    Event         *event;
{
    sub2_dncwin_objects    *ip = (sub2_dncwin_objects *) xv_get(item, XV_KEY_DATA, INS);

    switch(value) {
    case 0 :
        strcpy(send_data.databit, "CS7");
        break;
    case 1 :
        strcpy(send_data.databit, "CS8");
        break;
    }

    fprintf(stderr, "databit: %s\n", send_data.databit);
}

/*
 * Notify callback function for `stopbit_set'.
 */
void
stopbit_proc(item, value, event)
    Panel_item    item;
    int           value;
    Event         *event;
{
    sub2_dncwin_objects    *ip = (sub2_dncwin_objects *) xv_get(item, XV_KEY_DATA, INS);

    switch(value) {
    case 0 :
        strcpy(send_data.stopbit, "CSTOPB");
        break;
    case 1 :
        strcpy(send_data.stopbit, "~CSTOPB");
        break;
    }

    fprintf(stderr, "stopbit: %s\n", send_data.stopbit);
}

/*
 * Notify callback function for `parity_set'.
 */
void
parity_proc(item, value, event)
    Panel_item    item;
    int           value;
    Event         *event;
{
    sub2_dncwin_objects    *ip = (sub2_dncwin_objects *) xv_get(item, XV_KEY_DATA, INS);

    switch(value) {
    case 0 :
        strcpy(send_data.parity, "NULL");
        break;
    case 1 :
        strcpy(send_data.parity, "PARENB");
        break;
    case 2 :
        strcpy(send_data.parity, "PARODD");
        break;
    }
}

```

```

    }
    fprintf(stderr, "parity: %s\n", send_data.parity);
}

/*
 * Program for Data Send to NC Machine
 */
void
send_nc()
{
    char    directory[BUFSIZE];
    char    buff[BUFSIZE];

    char    c;
    int     fdl, i, p_value, pre_value = 0;
    long    cur_pos, fsize;

    FILE    *fp;

    xv_set(sub2_dncwin->slider1, PANEL_VALUE, 0, NULL);

    fdl = sio_open();
    if(fdl < 0){
        notice_prompt(sub2_dncwin->machine_control, NULL,
            NOTICE_MESSAGE_STRINGS, "   Error 발생 !",
            "기계가 연결되지 않음", NULL,
            NOTICE_BUTTON_YES, "확인",
            NULL);
        return;
    }
    printf("\n Open OK !");

    sprintf(directory, "%s%s%s", ROOTDIR, p_name, det_cadm);
    sprintf(buff, "%s/%s", directory, current_value);
    printf("\n File : %s", buff);

    if((fp = fopen(buff, "rb")) == NULL){
        notice_prompt(sub2_dncwin->machine_control, NULL,
            NOTICE_MESSAGE_STRINGS, "   Error 발생 !",
            "해당 파일을 열수 없음", NULL,
            NOTICE_BUTTON_YES, "확인",
            NULL);
        return;
    }

    cur_pos = ftell(fp);
    fseek(fp, 0L, SEEK_END);
    fsize = ftell(fp);
    fseek(fp, cur_pos, SEEK_SET);

    i = 0;
    while(1){
        if(feof(fp)){
            fclose(fp);
            printf("\n Send OK !\n");
            return;
        }
        c=fgetc(fp);

        write(fdl, &c, 1);

        p_value = (100*i++)/(int)fsize;
        if(p_value%10==0 && p_value!=pre_value) {
            xv_set(sub2_dncwin->slider1,
                PANEL_VALUE, p_value, NULL);
            pre_value = p_value;
        }
    }
}

```



```

    }
}

/*****
**** serial I/O open ****
*****/
int sio_open()
{
    struct termios new_tty;
    int fd;

    fd = open(send_data.machine_value, O_RDWR);
    if(fd < 0) return(-1);

    ioctl (fd, TCGETS, &new_tty);
    new_tty.c_iflag = 0;
    new_tty.c_oflag &= ~(OPOST);
    new_tty.c_lflag &= ~(ISIG | ICANON | ECHO | ECHOE | ECHOK | XCASE);

/*
    if(!strcmp(send_data.parity, "NULL"))
        new_tty.c_cflag = (send_data.baudrate | send_data.databit
                           | send_data.stopbit | send_data.parity
                           | HUPCL | CLOCAL | CREAD | CRTSCTS );
    else
        new_tty.c_cflag = (send_data.baudrate | send_data.databit | send_data.stopbit
                           | HUPCL | CLOCAL | CREAD | CRTSCTS );
*/
    new_tty.c_cflag = (B9600 | HUPCL | CLOCAL | CREAD | CS8 | CRTSCTS );

    new_tty.c_cc[4] = 1;
    new_tty.c_cc[5] = 0;

    ioctl (fd, TCSETS, &new_tty);
    return(fd);
}

/*
 * Program for Data Send to Satellite Computer
 */
void send_pc()
{
    notice_prompt(sub2_dncwin->machine_control, NULL,
        NOTICE_MESSAGE_STRINGS, " Error 발생 !",
        "PC가 연결되지 않음", NULL,
        NOTICE_BUTTON_YES, "확인",
        NULL);
}

/*
 * Program for Data Send to NC Tape Machine
 */
void send_tape()
{
    notice_prompt(sub2_dncwin->machine_control, NULL,
        NOTICE_MESSAGE_STRINGS, " Error 발생 !",
        "천공테이프가 연결되지 않음", NULL,
        NOTICE_BUTTON_YES, "확인",
        NULL);
}

void

```

```

send_proc()
{
    int    pid;

    if( !current_value[0] ) {
        notice_prompt(sub2_dncwin->data_sr_control, NULL,
            NOTICE_MESSAGE_STRINGS, "선택된 File이 없습니다",
            "File을 먼저 선택하십시오", NULL,
            NOTICE_BUTTON_YES, "확인",
            NULL);
        return;
    }

    switch(ncpc) {
    case 0 :
        send_nc();
/*
        pid = getpid();
        if( fork() == 0 ) {
            int status;
            execl("/usr3/hkim/menu/hyun/cim/send", "send", "/dev/ttya", current_value
);
            wait(&status);
        }
*/
        break;
    case 1 :
        send_pc();
        break;
    case 2 :
        send_tape();
        break;
    }
}

/*
 * Notify callback function for `button1`.
 */
void
send_procl(item, event)
    Panel_item    item;
    Event          *event;
{
    sub2_dncwin_objects *ip = (sub2_dncwin_objects *) xv_get(item, XV_KEY_DATA, INS
);

    sprintf(send_data.machine_value, "%s", SERIAL_A);
    send_proc();
}

/*
 * Notify callback function for `button2`.
 */
void
send_proc2(item, event)
    Panel_item    item;
    Event          *event;
{
    sub2_dncwin_objects *ip = (sub2_dncwin_objects *) xv_get(item, XV_KEY_DATA, INS
);

    sprintf(send_data.machine_value, "%s", SERIAL_B);
    send_proc();
}

/*

```

```

    * Notify callback function for `button3`.
    */
void
send_proc3(item, event)
    Panel_item    item;
    Event         *event;
{
    sub2_dncwin_objects *ip = (sub2_dncwin_objects *) xv_get(item, XV_KEY_DATA, IN
);

    bzero(send_data.machine_value, BUFSIZE);
    send_proc();
}

/*
 * Notify callback function for `button4`.
 */
void
send_proc4(item, event)
    Panel_item    item;
    Event         *event;
{
    sub2_dncwin_objects *ip = (sub2_dncwin_objects *) xv_get(item, XV_KEY_DATA, IN
);

    bzero(send_data.machine_value, BUFSIZE);
    send_proc();
}

/*
 * Notify callback function for `button5`.
 */
void
send_proc5(item, event)
    Panel_item    item;
    Event         *event;
{
    sub2_dncwin_objects *ip = (sub2_dncwin_objects *) xv_get(item, XV_KEY_DATA, IN
);

    bzero(send_data.machine_value, BUFSIZE);
    send_proc();
}

/*
 * Notify callback function for `button6`.
 */
void
send_proc8(item, event)
    Panel_item    item;
    Event         *event;
{
    sub2_dncwin_objects *ip = (sub2_dncwin_objects *) xv_get(item, XV_KEY_DATA, IN
);

    bzero(send_data.machine_value, BUFSIZE);
    send_proc();
}

```

```

#include <stdio.h>
#include <sys/param.h>
#include <sys/types.h>
#include <xview/xview.h>
#include <xview/canvas.h>
#include <xview/icon_load.h>
#include <xview/panel.h>
#include <xview/scrollbar.h>
#include <xview/svimage.h>
#include <xview/termsw.h>
#include <xview/text.h>
#include <xview/tty.h>
#include <xview/xv_xrect.h>
#include <gcm.h>
#include "sub2_ui.h"

/*
 * Initialize an instance of object `dncwin'.
 */
sub2_dncwin_objects *
sub2_dncwin_objects_initialize(ip, owner)
    sub2_dncwin_objects *ip;
    Xv_opaque owner;
{
    if (!ip && !(ip = (sub2_dncwin_objects *) calloc(1, sizeof (sub2_dncwin_objects))))
        return (sub2_dncwin_objects *) NULL;
    if (!ip->dncwin)
        ip->dncwin = sub2_dncwin_dncwin_create(ip, owner);
    if (!ip->pop_control)
        ip->pop_control = sub2_dncwin_pop_control_create(ip, ip->dncwin);
    if (!ip->pop_control_msg)
        ip->pop_control_msg = sub2_dncwin_pop_control_msg_create(ip, ip->pop_control);
    if (!ip->subl_control)
        ip->subl_control = sub2_dncwin_subl_control_create(ip, ip->dncwin);
    if (!ip->send_msg)
        ip->send_msg = sub2_dncwin_send_msg_create(ip, ip->subl_control);
    if (!ip->send_type1)
        ip->send_type1 = sub2_dncwin_send_type1_create(ip, ip->subl_control);
    if (!ip->data_sr_control)
        ip->data_sr_control = sub2_dncwin_data_sr_control_create(ip, ip->dncwin);
    if (!ip->send_choice_msg)
        ip->send_choice_msg = sub2_dncwin_send_choice_msg_create(ip, ip->data_sr_control);
);
    if (!ip->baudrate_set)
        ip->baudrate_set = sub2_dncwin_baudrate_set_create(ip, ip->data_sr_control);
    if (!ip->send_condition_msg)
        ip->send_condition_msg = sub2_dncwin_send_condition_msg_create(ip, ip->data_sr_control);
    if (!ip->databit_set)
        ip->databit_set = sub2_dncwin_databit_set_create(ip, ip->data_sr_control);
    if (!ip->stopbit_set)
        ip->stopbit_set = sub2_dncwin_stopbit_set_create(ip, ip->data_sr_control);
    if (!ip->parity_set)
        ip->parity_set = sub2_dncwin_parity_set_create(ip, ip->data_sr_control);
    if (!ip->list_control1)
        ip->list_control1 = sub2_dncwin_list_control1_create(ip, ip->dncwin);
    if (!ip->product_list)
        ip->product_list = sub2_dncwin_product_list_create(ip, ip->list_control1);
    if (!ip->list_control2)
        ip->list_control2 = sub2_dncwin_list_control2_create(ip, ip->dncwin);
    if (!ip->filelist)
        ip->filelist = sub2_dncwin_filelist_create(ip, ip->list_control2);
    if (!ip->file_editor)
        ip->file_editor = sub2_dncwin_file_editor_create(ip, ip->dncwin);
    if (!ip->machine_control)
        ip->machine_control = sub2_dncwin_machine_control_create(ip, ip->dncwin);
    if (!ip->button1)
        ip->button1 = sub2_dncwin_button1_create(ip, ip->machine_control);
}

```



```

        if (!ip->machine1_name_msg)
            ip->machine1_name_msg = sub2_dncwin_machine1_name_msg_create(ip, ip->machine_control);
    trol);
    if (!ip->slider1)
        ip->slider1 = sub2_dncwin_slider1_create(ip, ip->machine_control);
    if (!ip->button2)
        ip->button2 = sub2_dncwin_button2_create(ip, ip->machine_control);
    if (!ip->machine2_name_msg)
        ip->machine2_name_msg = sub2_dncwin_machine2_name_msg_create(ip, ip->machine_control);
    trol);
    if (!ip->slider2)
        ip->slider2 = sub2_dncwin_slider2_create(ip, ip->machine_control);
    if (!ip->button3)
        ip->button3 = sub2_dncwin_button3_create(ip, ip->machine_control);
    if (!ip->machine3_name_msg)
        ip->machine3_name_msg = sub2_dncwin_machine3_name_msg_create(ip, ip->machine_control);
    trol);
    if (!ip->slider3)
        ip->slider3 = sub2_dncwin_slider3_create(ip, ip->machine_control);
    if (!ip->button4)
        ip->button4 = sub2_dncwin_button4_create(ip, ip->machine_control);
    if (!ip->machine4_name_msg)
        ip->machine4_name_msg = sub2_dncwin_machine4_name_msg_create(ip, ip->machine_control);
    trol);
    if (!ip->slider4)
        ip->slider4 = sub2_dncwin_slider4_create(ip, ip->machine_control);
    if (!ip->button5)
        ip->button5 = sub2_dncwin_button5_create(ip, ip->machine_control);
    if (!ip->machine5_name_msg)
        ip->machine5_name_msg = sub2_dncwin_machine5_name_msg_create(ip, ip->machine_control);
    trol);
    if (!ip->slider5)
        ip->slider5 = sub2_dncwin_slider5_create(ip, ip->machine_control);
    if (!ip->button6)
        ip->button6 = sub2_dncwin_button6_create(ip, ip->machine_control);
    if (!ip->machine6_name_msg)
        ip->machine6_name_msg = sub2_dncwin_machine6_name_msg_create(ip, ip->machine_control);
    trol);
    if (!ip->slider6)
        ip->slider6 = sub2_dncwin_slider6_create(ip, ip->machine_control);
    return ip;
}

/*
 * Create object `dncwin' in the specified instance.
 */
Xv_opaque
sub2_dncwin_dncwin_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, FRAME_CMD,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_WIDTH, 1140,
                    XV_HEIGHT, 753,
                    XV_SHOW, FALSE,
                    FRAME_SHOW_FOOTER, FALSE,
                    FRAME_SHOW_RESIZE_CORNER, FALSE,
                    FRAME_CMD_PUSHPIN_IN, TRUE,
                    NULL);
    gcm_initialize_colors(obj, "Black", "Yellow");
    xv_set(xv_get(obj, FRAME_CMD_PANEL), WIN_SHOW, FALSE, NULL);
    return obj;
}

```



```

/*
 * Create object `pop_control' in the specified instance.
 */
Xv_opaque
sub2_dncwin_pop_control_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 0,
                    XV_Y, 0,
                    XV_WIDTH, 150,
                    XV_HEIGHT, 100,
                    WIN_BORDER, TRUE,
                    NULL);
    gcm_initialize_colors(obj, "Dark Slate Gray", "Yellow")
    return obj;
}

/*
 * Create object `pop_control_msg' in the specified instance.
 */
Xv_opaque
sub2_dncwin_pop_control_msg_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;
    Xv_opaque    pop_control_msg_image;
    static unsigned short pop_control_msg_bits[] = {
#include "dnc.icon"
    };

    pop_control_msg_image = xv_create(XV_NULL, SERVER_IMAGE,
                                      SERVER_IMAGE_BITS, pop_control_msg_bits,
                                      SERVER_IMAGE_DEPTH, 1,
                                      XV_WIDTH, 64,
                                      XV_HEIGHT, 64,
                                      NULL);
    obj = xv_create(owner, PANEL_MESSAGE,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 36,
                    XV_Y, 20,
                    XV_WIDTH, 64,
                    XV_HEIGHT, 64,
                    PANEL_ITEM_COLOR, gcm_color_index("Cyan"),
                    PANEL_LABEL_IMAGE, pop_control_msg_image,
                    PANEL_LABEL_BOLD, FALSE,
                    NULL);
    return obj;
}

/*
 * Create object `sub1_control' in the specified instance.
 */
Xv_opaque
sub2_dncwin_sub1_control_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

```

```

    obj = xv_create(owner, PANEL,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 150,
        XV_Y, 0,
        XV_WIDTH, 620,
        XV_HEIGHT, 100,
        WIN_BORDER, TRUE,
        NULL);
    gcm_initialize_colors(obj, "Dark Slate Gray", "White");
    return obj;
}

/*
 * Create object `send_msg' in the specified instance.
 */
Xv_opaque
sub2_dncwin_send_msg_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_MESSAGE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 28,
        XV_Y, 24,
        XV_WIDTH, 281,
        XV_HEIGHT, 16,
        PANEL_ITEM_COLOR, gcm_color_index("Yellow"),
        PANEL_LABEL_STRING, "RS-232C에 의한 가공 기계로의 데이터 전송",
        PANEL_LABEL_BOLD, TRUE,
        NULL);
    return obj;
}

/*
 * Create object `send_type1' in the specified instance.
 */
Xv_opaque
sub2_dncwin_send_type1_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern int    sendtype_proc();
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_CHOICE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 28,
        XV_Y, 52,
        XV_WIDTH, 422,
        XV_HEIGHT, 26,
        PANEL_VALUE_X, 100,
        PANEL_VALUE_Y, 52,
        PANEL_CHOICE_NROWS, 1,
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_ITEM_COLOR, gcm_color_index("Yellow"),
        PANEL_LABEL_STRING, "전송방법:",
        PANEL_NOTIFY_PROC, sendtype_proc,
        PANEL_CHOICE_STRINGS,
            " NC로 직접 전송 ",
            "PC로 Download ",
            "NC 천공 테이프 ",
            0,
        PANEL_CHOICE_COLOR, 0, gcm_color_index("Blue"),
        PANEL_CHOICE_COLOR, 1, gcm_color_index("Blue"));
}

```

```

                PANEL_CHOICE_COLOR, 2, gcm_color_index("Blue"),
                NULL);
        return obj;
    }

/*
 * Create object `data_sr_control' in the specified instance.
 */
Xv_opaque
sub2_dncwin_data_sr_control_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 771,
                    XV_Y, 0,
                    XV_WIDTH, WIN_EXTEND_TO_EDGE,
                    XV_HEIGHT, 179,
                    WIN_BORDER, TRUE,
                    NULL);
    gcm_initialize_colors(obj, "Black", NULL);
    return obj;
}

/*
 * Create object `send_choice_msg' in the specified instance.
 */
Xv_opaque
sub2_dncwin_send_choice_msg_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_MESSAGE,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 24,
                    XV_Y, 20,
                    XV_WIDTH, 148,
                    XV_HEIGHT, 18,
                    PANEL_ITEM_COLOR, gcm_color_index("Yellow"),
                    PANEL_LABEL_STRING, "<< 데이터 전송조건 >>",
                    PANEL_LABEL_BOLD, TRUE,
                    NULL);
    return obj;
}

/*
 * Create object `baudrate_set' in the specified instance.
 */
Xv_opaque
sub2_dncwin_baudrate_set_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern int      baudrate_proc();
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_CHOICE, PANEL_DISPLAY_LEVEL, PANEL_CURRENT,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 160,
                    XV_Y, 48,

```

```

        XV_WIDTH, 150,
        XV_HEIGHT, 26,
        PANEL_VALUE_X, 239,
        PANEL_VALUE_Y, 48,
        PANEL_CHOICE_NROWS, 1,
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_ITEM_COLOR, gcm_color_index("Yellow"),
        PANEL_LABEL_STRING, "Baudrate :",
        PANEL_NOTIFY_PROC, baudrate_proc,
        PANEL_CHOICE_STRINGS,
            "9600",
            "300",
            "1200",
            "2400",
            "4800",
            "19200",
            0,
        PANEL_CHOICE_COLOR, 0, gcm_color_index("red"),
        PANEL_CHOICE_COLOR, 1, gcm_color_index("Yellow"),
        PANEL_CHOICE_COLOR, 2, gcm_color_index("Yellow"),
        PANEL_CHOICE_COLOR, 3, gcm_color_index("Yellow"),
        PANEL_CHOICE_COLOR, 4, gcm_color_index("Yellow"),
        PANEL_CHOICE_COLOR, 5, gcm_color_index("Yellow"),
        NULL);
    return obj;
}

/*
 * Create object `send_condition_msg' in the specified instance.
 */
Xv_opaque
sub2_dncwin_send_condition_msg_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;
    Xv_opaque    send_condition_msg_image;
    static unsigned short send_condition_msg_bits[] = {
#include "machine_msg.icon"
    };

    send_condition_msg_image = xv_create(XV_NULL, SERVER_IMAGE
        SERVER_IMAGE_BITS, send_condition_msg_bits,
        SERVER_IMAGE_DEPTH, 1,
        XV_WIDTH, 64,
        XV_HEIGHT, 64,
        NULL);
    obj = xv_create(owner, PANEL_MESSAGE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 52,
        XV_Y, 68,
        XV_WIDTH, 64,
        XV_HEIGHT, 64,
        PANEL_ITEM_COLOR, gcm_color_index("Yellow"),
        PANEL_LABEL_IMAGE, send_condition_msg_image,
        PANEL_LABEL_BOLD, TRUE,
        NULL);
    return obj;
}

/*
 * Create object `databit_set' in the specified instance.
 */
Xv_opaque
sub2_dncwin_databit_set_create(ip, owner)
    caddr_t      ip;

```

```

    Xv_opaque      owner;
}

extern int          databit_proc();
Xv_opaque          obj;

obj = xv_create(owner, PANEL_CHOICE, PANEL_DISPLAY_LEVEL, PANEL_CURRENT
    XV_KEY_DATA, INSTANCE, ip,
    XV_X, 160,
    XV_Y, 76,
    XV_WIDTH, 129,
    XV_HEIGHT, 26,
    PANEL_VALUE_X, 239,
    PANEL_VALUE_Y, 76,
    PANEL_CHOICE_NROWS, 1,
    PANEL_LAYOUT, PANEL_HORIZONTAL,
    PANEL_ITEM_COLOR, gcm_color_index("Yellow"),
    PANEL_LABEL_STRING, "Data Bit :",
    PANEL_NOTIFY_PROC, databit_proc,
    PANEL_CHOICE_STRINGS,
        "8 ",
        "7 ",
        0,
    PANEL_CHOICE_COLOR, 0, gcm_color_index("Red"),
    PANEL_CHOICE_COLOR, 1, gcm_color_index("Yellow"),
    NULL);
return obj;
}

/*
 * Create object `stopbit_set' in the specified instance.
 */
Xv_opaque
sub2_dncwin_stopbit_set_create(ip, owner)
    caddr_t          ip;
    Xv_opaque        owner;
{
    extern int          stopbit_proc();
    Xv_opaque          obj;

    obj = xv_create(owner, PANEL_CHOICE, PANEL_DISPLAY_LEVEL, PANEL_CURRENT
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 160,
        XV_Y, 104,
        XV_WIDTH, 143,
        XV_HEIGHT, 26,
        PANEL_VALUE_X, 239,
        PANEL_VALUE_Y, 104,
        PANEL_CHOICE_NROWS, 1,
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_ITEM_COLOR, gcm_color_index("Yellow"),
        PANEL_LABEL_STRING, "Stop Bit :",
        PANEL_NOTIFY_PROC, stopbit_proc,
        PANEL_CHOICE_STRINGS,
            "OFF ",
            "ON ",
            0,
        PANEL_CHOICE_COLOR, 0, gcm_color_index("Red"),
        PANEL_CHOICE_COLOR, 1, gcm_color_index("Yellow"),
        NULL);
    return obj;
}

/*
 * Create object `parity_set' in the specified instance.
 */
Xv_opaque

```



```

sub2_dncwin_parity_set_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern int      parity_proc();
    Xv_opaque      obj;

    obj = xv_create(owner, PANEL_CHOICE, PANEL_DISPLAY_LEVEL, PANEL_CURRENT
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 160,
        XV_Y, 132,
        XV_WIDTH, 150,
        XV_HEIGHT, 26,
        PANEL_VALUE_X, 239,
        PANEL_VALUE_Y, 132,
        PANEL_CHOICE_NROWS, 1,
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_ITEM_COLOR, gcm_color_index("Yellow"),
        PANEL_LABEL_STRING, "Parity  :",
        PANEL_NOTIFY_PROC, parity_proc,
        PANEL_CHOICE_STRINGS,
            "NONE",
            "Even ",
            "Odd  ",
            0,
        PANEL_CHOICE_COLOR, 0, gcm_color_index("Red"),
        PANEL_CHOICE_COLOR, 1, gcm_color_index("Yellow"),
        PANEL_CHOICE_COLOR, 2, gcm_color_index("Yellow"),
        NULL);
    return obj;
}

/*
 * Create object `list_controll' in the specified instance.
 */
Xv_opaque
sub2_dncwin_list_controll_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque      obj;

    obj = xv_create(owner, PANEL,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 0,
        XV_Y, 100,
        XV_WIDTH, 150,
        XV_HEIGHT, WIN_EXTEND_TO_EDGE,
        WIN_BORDER, TRUE,
        NULL);
    gcm_initialize_colors(obj, "Dark Slate Gray", "White");
    return obj;
}

/*
 * Create object `product_list' in the specified instance.
 */
Xv_opaque
sub2_dncwin_product_list_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern int      dnc_plist_proc();
    Xv_opaque      obj;

    obj = xv_create(owner, PANEL_LIST,

```

```

        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 4,
        XV_Y, 4,
        PANEL_LIST_WIDTH, 125,
        XV_HEIGHT, 649,
        PANEL_LIST_DISPLAY_ROWS, 29,
        PANEL_ITEM_COLOR, gcm_color_index("Medium Spring Green"),
        PANEL_LABEL_STRING, " 제품명 : ",
        PANEL_LAYOUT, PANEL_VERTICAL,
        PANEL_READ_ONLY, FALSE,
        PANEL_CHOOSE_ONE, TRUE,
        PANEL_CHOOSE_NONE, TRUE,
        PANEL_NOTIFY_PROC, dnc_plist_proc,
        NULL);
    return obj;
}

/*
 * Create object `list_control2' in the specified instance.
 */
Xv_opaque
sub2_dncwin_list_control2_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 150,
        XV_Y, 100,
        XV_WIDTH, 310,
        XV_HEIGHT, WIN_EXTEND_TO_EDGE,
        WIN_BORDER, TRUE,
        NULL);
    gcm_initialize_colors(obj, "Dark Slate Gray", "Yellow");
    return obj;
}

/*
 * Create object `filelist' in the specified instance.
 */
Xv_opaque
sub2_dncwin_filelist_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern int    dnc_flist_proc();
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_LIST,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 4,
        XV_Y, 4,
        PANEL_LIST_WIDTH, 283,
        XV_HEIGHT, 649,
        PANEL_LIST_DISPLAY_ROWS, 29,
        PANEL_ITEM_COLOR, gcm_color_index("White"),
        PANEL_LABEL_STRING, " 순번      파일명      크기      설계자",
        PANEL_LAYOUT, PANEL_VERTICAL,
        PANEL_READ_ONLY, FALSE,
        PANEL_CHOOSE_ONE, TRUE,
        PANEL_CHOOSE_NONE, TRUE,
        PANEL_NOTIFY_PROC, dnc_flist_proc,
        NULL);
    return obj;
}

```

```

}

/*
 * Create object `file_editor' in the specified instance.
 */
Xv_opaque
sub2_dncwin_file_editor_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, TEXTSW,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 464,
                    XV_Y, 125,
                    XV_WIDTH, 305,
                    XV_HEIGHT, WIN_EXTEND_TO_EDGE,
                    OPENWIN_SHOW_BORDERS, TRUE,
                    NULL);
    gcm_initialize_colors(obj, "Black", "White");
    return obj;
}

/*
 * Create object `machine_control' in the specified instance
 */
Xv_opaque
sub2_dncwin_machine_control_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 770,
                    XV_Y, 180,
                    XV_WIDTH, WIN_EXTEND_TO_EDGE,
                    XV_HEIGHT, WIN_EXTEND_TO_EDGE,
                    WIN_BORDER, TRUE,
                    NULL);
    gcm_initialize_colors(obj, "Cornflower Blue", NULL);
    return obj;
}

/*
 * Create object `button1' in the specified instance.
 */
Xv_opaque
sub2_dncwin_button1_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern void      send_procl();
    Xv_opaque    obj;
    Xv_opaque    button1_image;
    static unsigned short button1_bits[] = {
#include "machine.icon"
    };

    button1_image = xv_create(XV_NULL, SERVER_IMAGE,
                              SERVER_IMAGE_BITS, button1_bits,
                              SERVER_IMAGE_DEPTH, 1,
                              XV_WIDTH, 64,

```

```

        XV_HEIGHT, 64,
        NULL);
    obj = xv_create(owner, PANEL_BUTTON,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 20,
        XV_Y, 24,
        XV_WIDTH, 86,
        XV_HEIGHT, 74,
        PANEL_ITEM_COLOR, gcm_color_index("Red"),
        PANEL_LABEL_IMAGE, button1_image,
        PANEL_NOTIFY_PROC, send_procl,
        NULL);
    return obj;
}

/*
 * Create object `machinel_name_msg' in the specified instance
 */
Xv_opaque
sub2_dncwin_machinel_name_msg_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_MESSAGE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 120,
        XV_Y, 32,
        XV_WIDTH, 141,
        XV_HEIGHT, 16,
        PANEL_ITEM_COLOR, gcm_color_index("Red"),
        PANEL_LABEL_STRING, "Milling Machine TTYA",
        PANEL_LABEL_BOLD, TRUE,
        NULL);
    return obj;
}

/*
 * Create object `slider1' in the specified instance.
 */
Xv_opaque
sub2_dncwin_slider1_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_GAUGE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 120,
        XV_Y, 68,
        XV_WIDTH, 212,
        XV_HEIGHT, 34,
        PANEL_VALUE_X, 192,
        PANEL_VALUE_Y, 68,
        PANEL_GAUGE_WIDTH, 120,
        PANEL_TICKS, 11,
        PANEL_ITEM_COLOR, gcm_color_index("Blue"),
        PANEL_LABEL_STRING, "작업진척:",
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_DIRECTION, PANEL_HORIZONTAL,
        PANEL_SHOW_RANGE, TRUE,
        PANEL_MIN_VALUE, 0,
        PANEL_MAX_VALUE, 100,
        NULL);
}

```

```

        return obj;
    }

/*
 * Create object `button2' in the specified instance.
 */
Xv_opaque
sub2_dncwin_button2_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern void      send_proc2();
    Xv_opaque      obj;
    Xv_opaque      button2_image;
    static unsigned short button2_bits[] = {
#include "machine.icon"
    };

    button2_image = xv_create(XV_NULL, SERVER_IMAGE,
        SERVER_IMAGE_BITS, button2_bits,
        SERVER_IMAGE_DEPTH, 1,
        XV_WIDTH, 64,
        XV_HEIGHT, 64,
        NULL);
    obj = xv_create(owner, PANEL_BUTTON,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 20,
        XV_Y, 112,
        XV_WIDTH, 86,
        XV_HEIGHT, 74,
        PANEL_ITEM_COLOR, gcm_color_index("Red"),
        PANEL_LABEL_IMAGE, button2_image,
        PANEL_NOTIFY_PROC, send_proc2,
        NULL);
    return obj;
}

/*
 * Create object `machine2_name_msg' in the specified instance
 */
Xv_opaque
sub2_dncwin_machine2_name_msg_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque      obj;

    obj = xv_create(owner, PANEL_MESSAGE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 120,
        XV_Y, 120,
        XV_WIDTH, 141,
        XV_HEIGHT, 16,
        PANEL_ITEM_COLOR, gcm_color_index("Red"),
        PANEL_LABEL_STRING, "Milling Machine TTYB",
        PANEL_LABEL_BOLD, TRUE,
        NULL);
    return obj;
}

/*
 * Create object `slider2' in the specified instance.
 */
Xv_opaque
sub2_dncwin_slider2_create(ip, owner)

```



```

    caddr_t      ip;
    Xv_opaque    owner;
}

Xv_opaque      obj;

obj = xv_create(owner, PANEL_GAUGE,
    XV_KEY_DATA, INSTANCE, ip,
    XV_X, 120,
    XV_Y, 144,
    XV_WIDTH, 212,
    XV_HEIGHT, 34,
    PANEL_VALUE_X, 192,
    PANEL_VALUE_Y, 144,
    PANEL_GAUGE_WIDTH, 120,
    PANEL_TICKS, 11,
    PANEL_ITEM_COLOR, gcm_color_index("Blue"),
    PANEL_LABEL_STRING, "작업진척:",
    PANEL_LAYOUT, PANEL_HORIZONTAL,
    PANEL_DIRECTION, PANEL_HORIZONTAL,
    PANEL_SHOW_RANGE, TRUE,
    PANEL_MIN_VALUE, 0,
    PANEL_MAX_VALUE, 100,
    NULL);
    return obj;
}

/*
 * Create object `button3' in the specified instance.
 */
Xv_opaque
sub2_dncwin_button3_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern void      send_proc3();
    Xv_opaque      obj;
    Xv_opaque      button3_image;
    static unsigned short button3_bits[] = {
#include "machine.icon"
    };

    button3_image = xv_create(XV_NULL, SERVER_IMAGE,
        SERVER_IMAGE_BITS, button3_bits,
        SERVER_IMAGE_DEPTH, 1,
        XV_WIDTH, 64,
        XV_HEIGHT, 64,
        NULL);
    obj = xv_create(owner, PANEL_BUTTON,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 20,
        XV_Y, 200,
        XV_WIDTH, 86,
        XV_HEIGHT, 74,
        PANEL_ITEM_COLOR, gcm_color_index("Black"),
        PANEL_LABEL_IMAGE, button3_image,
        PANEL_NOTIFY_PROC, send_proc3,
        NULL);
    return obj;
}

/*
 * Create object `machine3_name_msg' in the specified instance
 */
Xv_opaque
sub2_dncwin_machine3_name_msg_create(ip, owner)
    caddr_t      ip;

```

```

    Xv_opaque    owner:
}
    Xv_opaque    obj:

    obj = xv_create(owner, PANEL_MESSAGE,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 120,
                    XV_Y, 208,
                    XV_WIDTH, 134,
                    XV_HEIGHT, 16,
                    PANEL_ITEM_COLOR, gcm_color_index("Black"),
                    PANEL_LABEL_STRING, "Milling Machine XXX",
                    PANEL_LABEL_BOLD, TRUE,
                    NULL);
    return obj;
}

/*
 * Create object `slider3' in the specified instance.
 */
Xv_opaque
sub2_dncwin_slider3_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner:
{
    Xv_opaque    obj:

    obj = xv_create(owner, PANEL_GAUGE,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 120,
                    XV_Y, 236,
                    XV_WIDTH, 212,
                    XV_HEIGHT, 34,
                    PANEL_VALUE_X, 192,
                    PANEL_VALUE_Y, 236,
                    PANEL_GAUGE_WIDTH, 120,
                    PANEL_TICKS, 11,
                    PANEL_ITEM_COLOR, gcm_color_index("Black"),
                    PANEL_LABEL_STRING, "작업진척:",
                    PANEL_LAYOUT, PANEL_HORIZONTAL,
                    PANEL_DIRECTION, PANEL_HORIZONTAL,
                    PANEL_SHOW_RANGE, TRUE,
                    PANEL_MIN_VALUE, 0,
                    PANEL_MAX_VALUE, 100,
                    NULL);
    return obj;
}

/*
 * Create object `button4' in the specified instance.
 */
Xv_opaque
sub2_dncwin_button4_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner:
{
    extern void    send_proc4();
    Xv_opaque      obj:
    Xv_opaque      button4_image;
    static unsigned short button4_bits[] = {
#include "machine.icon"
    };

    button4_image = xv_create(XV_NULL, SERVER_IMAGE,
                              SERVER_IMAGE_BITS, button4_bits,
                              SERVER_IMAGE_DEPTH, 1,

```

```

        XV_WIDTH, 64,
        XV_HEIGHT, 64,
        NULL);
    obj = xv_create(owner, PANEL_BUTTON,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 20,
        XV_Y, 288,
        XV_WIDTH, 86,
        XV_HEIGHT, 74,
        PANEL_ITEM_COLOR, gcm_color_index("Black"),
        PANEL_LABEL_IMAGE, button4_image,
        PANEL_NOTIFY_PROC, send_proc4,
        NULL);
    return obj;
}

/*
 * Create object `machine4_name_msg' in the specified instance
 */
Xv_opaque
sub2_dncwin_machine4_name_msg_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_MESSAGE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 120,
        XV_Y, 296,
        XV_WIDTH, 134,
        XV_HEIGHT, 16,
        PANEL_ITEM_COLOR, gcm_color_index("Black"),
        PANEL_LABEL_STRING, "Milling Machine XXX",
        PANEL_LABEL_BOLD, TRUE,
        NULL);
    return obj;
}

/*
 * Create object `slider4' in the specified instance.
 */
Xv_opaque
sub2_dncwin_slider4_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_GAUGE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 120,
        XV_Y, 320,
        XV_WIDTH, 212,
        XV_HEIGHT, 34,
        PANEL_VALUE_X, 192,
        PANEL_VALUE_Y, 320,
        PANEL_GAUGE_WIDTH, 120,
        PANEL_TICKS, 11,
        PANEL_ITEM_COLOR, gcm_color_index("Black"),
        PANEL_LABEL_STRING, "작업진척:",
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_DIRECTION, PANEL_HORIZONTAL,
        PANEL_SHOW_RANGE, TRUE,
        PANEL_MIN_VALUE, 0,
        PANEL_MAX_VALUE, 100,

```

```

        NULL);
    return obj;
}

/*
 * Create object `button5' in the specified instance.
 */
Xv_opaque
sub2_dncwin_button5_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern void      send_proc5();
    Xv_opaque      obj;
    Xv_opaque      button5_image;
    static unsigned short button5_bits[] = {
#include "machine.icon"
    };

    button5_image = xv_create(XV_NULL, SERVER_IMAGE,
        SERVER_IMAGE_BITS, button5_bits,
        SERVER_IMAGE_DEPTH, 1,
        XV_WIDTH, 64,
        XV_HEIGHT, 64,
        NULL);
    obj = xv_create(owner, PANEL_BUTTON,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 20,
        XV_Y, 380,
        XV_WIDTH, 86,
        XV_HEIGHT, 74,
        PANEL_ITEM_COLOR, gcm_color_index("Black"),
        PANEL_LABEL_IMAGE, button5_image,
        PANEL_NOTIFY_PROC, send_proc5,
        NULL);
    return obj;
}

/*
 * Create object `machine5_name_msg' in the specified instance
 */
Xv_opaque
sub2_dncwin_machine5_name_msg_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque      obj;

    obj = xv_create(owner, PANEL_MESSAGE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 120,
        XV_Y, 388,
        XV_WIDTH, 134,
        XV_HEIGHT, 16,
        PANEL_ITEM_COLOR, gcm_color_index("Black"),
        PANEL_LABEL_STRING, "Milling Machine XXX",
        PANEL_LABEL_BOLD, TRUE,
        NULL);
    return obj;
}

/*
 * Create object `slider5' in the specified instance.
 */
Xv_opaque

```

```

sub2_dncwin_slider5_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_GAUGE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 120,
        XV_Y, 420,
        XV_WIDTH, 212,
        XV_HEIGHT, 34,
        PANEL_VALUE_X, 192,
        PANEL_VALUE_Y, 420,
        PANEL_GAUGE_WIDTH, 120,
        PANEL_TICKS, 11,
        PANEL_ITEM_COLOR, gcm_color_index("Black"),
        PANEL_LABEL_STRING, "작업진척:",
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_DIRECTION, PANEL_HORIZONTAL,
        PANEL_SHOW_RANGE, TRUE,
        PANEL_MIN_VALUE, 0,
        PANEL_MAX_VALUE, 100,
        NULL);
    return obj;
}

/*
 * Create object `button6' in the specified instance.
 */
Xv_opaque
sub2_dncwin_button6_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern void      send_proc6();
    Xv_opaque        obj;
    Xv_opaque        button6_image;
    static unsigned short button6_bits[] = {
#include "machine.icon"
    };

    button6_image = xv_create(XV_NULL, SERVER_IMAGE,
        SERVER_IMAGE_BITS, button6_bits,
        SERVER_IMAGE_DEPTH, 1,
        XV_WIDTH, 64,
        XV_HEIGHT, 64,
        NULL);
    obj = xv_create(owner, PANEL_BUTTON,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 20,
        XV_Y, 472,
        XV_WIDTH, 86,
        XV_HEIGHT, 74,
        PANEL_ITEM_COLOR, gcm_color_index("Black"),
        PANEL_LABEL_IMAGE, button6_image,
        PANEL_NOTIFY_PROC, send_proc6,
        NULL);
    return obj;
}

/*
 * Create object `machine6_name_msg' in the specified instance
 */
Xv_opaque
sub2_dncwin_machine6_name_msg_create(ip, owner)

```



```

        caddr_t      ip;
        Xv_opaque    owner;
    {
        Xv_opaque    obj;

        obj = xv_create(owner, PANEL_MESSAGE,
            XV_KEY_DATA, INSTANCE, ip,
            XV_X, 120,
            XV_Y, 480,
            XV_WIDTH, 134,
            XV_HEIGHT, 16,
            PANEL_ITEM_COLOR, gcm_color_index("Black"),
            PANEL_LABEL_STRING, "Milling Machine XXX",
            PANEL_LABEL_BOLD, TRUE,
            NULL);
        return obj;
    }

/*
 * Create object `slider6' in the specified instance.
 */
Xv_opaque
sub2_dncwin_slider6_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_GAUGE,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 120,
        XV_Y, 508,
        XV_WIDTH, 212,
        XV_HEIGHT, 34,
        PANEL_VALUE_X, 192,
        PANEL_VALUE_Y, 508,
        PANEL_GAUGE_WIDTH, 120,
        PANEL_TICKS, 11,
        PANEL_ITEM_COLOR, gcm_color_index("Black"),
        PANEL_LABEL_STRING, "작업진척:",
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_DIRECTION, PANEL_HORIZONTAL,
        PANEL_SHOW_RANGE, TRUE,
        PANEL_MIN_VALUE, 0,
        PANEL_MAX_VALUE, 100,
        NULL);
    return obj;
}

```

```

#include <stdio.h>
#include <sys/param.h>
#include <sys/types.h>
#include <xview/xview.h>
#include <xview/panel.h>
#include <xview/textsw.h>
#include <xview/xv_xrect.h>
#include <gdd.h>
#include "main.h"
#include "pop_ui.h"
#include "subl_ui.h"

subl_dwin_objects      *subl_dwin;

extern int      file_list();
extern void     flist_display();
extern char     directory[BUFSIZE];
extern char     current_value[BUFSIZE];

#ifdef MAIN

/*
 * Instance XV_KEY_DATA key.  An instance is a set of related
 * user interface objects.  A pointer to an object's instance
 * is stored under this key in every object.  This must be a
 * global variable.
 */
Attr_attribute  INSTANCE;

void
main(argc, argv)
    int          argc;
    char         **argv;
{
    pop_popup_objects      *pop_popup;

    /*
     * Initialize XView.
     */
    xv_init(XV_INIT_ARGC_PTR_ARGV, &argc, argv, 0);
    INSTANCE = xv_unique_key();

    /*
     * Initialize user interface components.
     */
    pop_popup = pop_popup_objects_initialize(NULL, NULL);

    /*
     * Turn control over to XView.
     */
    xv_main_loop(pop_popup->popup);
    exit(0);
}

#endif

/*
 * Notify callback function for `pop_textfield'.
 */
Panel_setting
pop_proc(item, event)
    Panel_item      item;
    Event           *event;
{
    pop_popup_objects      *ip = (pop_popup_objects *) xv_get(item, XV_KEY_DATA, INSTA
    char*      value = (char *) xv_get(item, PANEL_VALUE);

```

```

static char buff[BUFSIZE];

xv_set(ip->popup, XV_SHOW, FALSE, NULL);
xv_set(ip->pop_textfield, PANEL_VALUE, "", 0);

if(value) {
    sprintf(buff, "/usr/bin/mv %s/%s %s/%s",
            directory, current_value, directory, value);
    if( system(buff) == -1 ) {
        perror("execl");
        exit(0);
    }

    system("");
    file_list();
    flist_display();
}

return panel_text_notify(item, event);
}

/*
 * Notify callback function for `pop_ok_button'.
 */
void
pop_ok_proc(item, event)
    Panel_item    item;
    Event         *event;
{
    pop_popup_objects *ip = (pop_popup_objects *) xv_get(item, XV_KEY_DATA, INST

xv_set(ip->popup, XV_SHOW, FALSE, NULL);
xv_set(ip->pop_textfield, PANEL_VALUE, "", 0);
}

```

```

#include <stdio.h>
#include <sys/param.h>
#include <sys/types.h>
#include <xview/xview.h>
#include <xview/canvas.h>
#include <xview/icon_load.h>
#include <xview/panel.h>
#include <xview/scrollbar.h>
#include <xview/svrmage.h>
#include <xview/termsw.h>
#include <xview/text.h>
#include <xview/tty.h>
#include <xview/xv_xrect.h>
#include <gcm.h>
#include "pop_ui.h"

/*
 * Initialize an instance of object `popup'.
 */
pop_popup_objects *
pop_popup_objects_initialize(ip, owner)
    pop_popup_objects *ip;
    Xv_opaque owner;
{
    if (!ip && !(ip = (pop_popup_objects *) calloc(1, sizeof (pop_popup_objects))))
        return (pop_popup_objects *) NULL;
    if (!ip->popup)
        ip->popup = pop_popup_popup_create(ip, owner);
    if (!ip->controls1)
        ip->controls1 = pop_popup_controls1_create(ip, ip->popup);
    if (!ip->tittle_msg)
        ip->tittle_msg = pop_popup_tittle_msg_create(ip, ip->controls1);
    if (!ip->filename_msg)
        ip->filename_msg = pop_popup_filename_msg_create(ip, ip->controls1);
    if (!ip->pop_textfield)
        ip->pop_textfield = pop_popup_pop_textfield_create(ip, ip->controls1);
    if (!ip->pop_ok_button)
        ip->pop_ok_button = pop_popup_pop_ok_button_create(ip, ip->controls1);
    if (!ip->messagel)
        ip->messagel = pop_popup_messagel_create(ip, ip->controls1);
    if (!ip->pop_msg)
        ip->pop_msg = pop_popup_pop_msg_create(ip, ip->controls1);
    return ip;
}

/*
 * Create object `popup' in the specified instance.
 */
Xv_opaque
pop_popup_popup_create(ip, owner)
    caddr_t ip;
    Xv_opaque owner;
{
    Xv_opaque obj;

    obj = xv_create(owner, FRAME_CMD,
        XV_KEY_DATA, INSTANCE, ip,
        XV_WIDTH, 414,
        XV_HEIGHT, 104,
        XV_SHOW, FALSE,
        FRAME_SHOW_FOOTER, FALSE,
        FRAME_SHOW_RESIZE_CORNER, FALSE,
        FRAME_CMD_PUSHPIN_IN, FALSE,
        NULL);
    xv_set(xv_get(obj, FRAME_CMD_PANEL), WIN_SHOW, FALSE, NULL);
    return obj;
}

```

```

/*
 * Create object `controls1' in the specified instance.
 */
Xv_opaque
pop_popup_controls1_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 0,
                    XV_Y, 0,
                    XV_WIDTH, WIN_EXTEND_TO_EDGE,
                    XV_HEIGHT, WIN_EXTEND_TO_EDGE,
                    WIN_BORDER, FALSE,
                    NULL);
    gcm_initialize_colors(obj, "Cadet Blue", "Yellow")
    return obj;
}

/*
 * Create object `tittle_msg' in the specified instance.
 */
Xv_opaque
pop_popup_tittle_msg_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_MESSAGE,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 20,
                    XV_Y, 20,
                    XV_WIDTH, 50,
                    XV_HEIGHT, 16,
                    PANEL_ITEM_COLOR, gcm_color_index("Black")
                    PANEL_LABEL_STRING, "기존 : ",
                    PANEL_LABEL_BOLD, TRUE,
                    NULL);
    return obj;
}

/*
 * Create object `filename_msg' in the specified instance.
 */
Xv_opaque
pop_popup_filename_msg_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_MESSAGE,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 76,
                    XV_Y, 20,
                    XV_WIDTH, 1,
                    XV_HEIGHT, 16,
                    PANEL_ITEM_COLOR, gcm_color_index("Black")
                    PANEL_LABEL_BOLD, TRUE,
                    NULL);
}

```



```

        return obj;
    }

/*
 * Create object `pop_textfield' in the specified instance
 */
Xv_opaque
pop_popup_pop_textfield_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern Panel_setting    pop_proc();
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_TEXT,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 204,
        XV_Y, 20,
        XV_WIDTH, 170,
        XV_HEIGHT, 18,
        PANEL_VALUE_X, 269,
        PANEL_VALUE_Y, 20,
        PANEL_VALUE_DISPLAY_LENGTH, 15,
        PANEL_VALUE_STORED_LENGTH, 15,
        PANEL_ITEM_COLOR, gcm_color_index("White"),
        PANEL_LABEL_STRING, "변경 : ",
        PANEL_LAYOUT, PANEL_HORIZONTAL,
        PANEL_READ_ONLY, FALSE,
        PANEL_NOTIFY_PROC, pop_proc,
        NULL);
    return obj;
}

/*
 * Create object `pop_ok_button' in the specified instance
 */
Xv_opaque
pop_popup_pop_ok_button_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    extern void    pop_ok_proc();
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_BUTTON,
        XV_KEY_DATA, INSTANCE, ip,
        XV_X, 24,
        XV_Y, 64,
        XV_WIDTH, 71,
        XV_HEIGHT, 21,
        PANEL_ITEM_COLOR, gcm_color_index("Blue"),
        PANEL_LABEL_STRING, "취소 ",
        PANEL_NOTIFY_PROC, pop_ok_proc,
        NULL);
    return obj;
}

/*
 * Create object `messagel' in the specified instance.
 */
Xv_opaque
pop_popup_messagel_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{

```

```

Xv_opaque      obj;

obj = xv_create(owner, PANEL_MESSAGE,
                XV_KEY_DATA, INSTANCE, ip,
                XV_X, 152,
                XV_Y, 68,
                XV_WIDTH, 43,
                XV_HEIGHT, 16,
                PANEL_ITEM_COLOR, gcm_color_index("Yellow")
                PANEL_LABEL_STRING, "메세지",
                PANEL_LABEL_BOLD, TRUE,
                NULL);
return obj;
}

/*
 * Create object `pop_msg' in the specified instance.
 */
Xv_opaque
pop_popup_pop_msg_create(ip, owner)
    caddr_t      ip;
    Xv_opaque    owner;
{
    Xv_opaque    obj;

    obj = xv_create(owner, PANEL_MESSAGE,
                    XV_KEY_DATA, INSTANCE, ip,
                    XV_X, 216,
                    XV_Y, 68,
                    XV_WIDTH, 148,
                    XV_HEIGHT, 16,
                    PANEL_ITEM_COLOR, gcm_color_index("Yellow")
                    PANEL_LABEL_STRING, "입력 후 Enter 하시오 "
                    PANEL_LABEL_BOLD, TRUE,
                    NULL);
return obj;
}

```