경영정보학연구
제 4권 2호
1994년 12월

범용언어에 의한 응용 프로그램 재사용 시스템의 설계 및 구현

오 무 송[1] 김 형 태[2]

A Study on the Design and Implementation of an Application Program Reuse System based on common language

Software development of large scale program such as Operating System or Universitry Total Information System is lengthy and costly process. In order to reduce cost, time and risk, there is currentry general acceptance of the need for Software Reuse System during the whole development cycles. In this paper, (from a practical point of view), the problem of existing reuse system methodology is analyzed and an implement method of software reuse system is presented. Also using this method Application Program Reuse System(APRS) which supports all phase of software life cycles is designed and implemented. This application program reuse system which is based on common language is considerably shown to reduce communication Error of requirement specification between systems analyst and end-user.

1) 조선대학교 공과대학 컴퓨터공학과
2) 조선대학교 공업전문대학 전자계산과

Ⅰ. 서 론

소프트웨어의 수요의 급증과 사용자의 편의성을 강조하는 추세에 비추어 소프트웨어의 전문기술인력은 상대적으로 부족한 실정이며 신뢰성 있는 고품질의 소프트웨어를 생산하기 위해서는 기존의 개발 방법론으로는 해결하기가 어렵다. 고품질과 고신뢰성의 소프트웨어를 생산 하기위한 하나의 방법이 하드웨어의 부품생산 개념을 도입하여 소프트웨어도 하드웨어의 부품처럼 생산하여 재사용 한다는 재사용 소프트웨어 개발 방법론이다. 소프트웨어 재사용이란 기존에 개발되어 실행과 검증을 거쳐 고품질임이 판명된 소프트웨어를 재사용하여 소프트웨어의 질과 신뢰성을 향상하고 개발시간과 비용을 감소시켜 소프트웨어의 위기를 극복 하기위한 주요한 방법중의 하나로 각종 모듈을 재사용하거나 현존하는 프로그램을 재사용 하는 것을 말한다. 소프트웨어의 재사용은 신뢰성이 확보된 기존의 소프트웨어를 재사용 하므로 신뢰성이 확보된 상태이며, 원시코드를 직접 작성하지 않고 기존의 소프트웨어를 재사용하므로 60% 정도의 생산성 향상이 가능하다는 연구결과도 있다.[Grasso,1984 ; Matsumoto, 1986]

그러나 기존의 재사용 방법에도 문제점은 있는데 그것은 다음과 같다. 첫째, 기존의 연구는 원시코드나 설계정보의 재사용등이 있으나 가장 기본적인 문제점, 즉 사용자와 개발자간의 의사소통 수단이나 방법의 미비로 인하여 생기는 시스템 명세 작성시의 착오를 줄일 방안이 없다. 둘째, 소프트웨어 수명주기 전 단계중에서 일부분만을 지원하고 있다. 즉 원시코드의 재사용 부분 혹은 설계단계의 재사용처럼 수명주기의 일부만 재사용하며 전과정을 망라한 시스템이 없다. 셋째, 기존의 재사용에 관한 연구는 사용자만을 위한 것이거나 혹은 개발자를 위한 시스템이며 이들 두 그룹을 모두 지원하는 방안이 미비하다. 넷째, 기존에 개발된 재사용 부품은 사용자 인터페이스등의 특수한 일부분의 업무로 한정되어 있어서 사용자의 다양한 응용영역의 요구에 미치지 못하고 있다. 마지막으로 기존의 재사용 시스템은 구현 언어가 일반적인 사용자들이 많이 쓰고 있는 범용의 언어가 아니므로 재사용 하고자 한다면 사용자 측의 모든 시스템은 재구성 되어야 한다. 이러한 문제점들을 해결하는 방안으로 본 논문에서는 시스템 명세 작성시의 착오를 줄이기 위해서 사용자가 기존의 재사용 부품을 확인하는 재사용 부품 검색 시스템을 설계하고, 적용 업무영역을 대학업무로 하며, 대학 구성원중 개발자와 사용자가 모두 공용하는 응용 프로그램 재사용 시스템(APRS : Application Program Reuse System)을 설계하고, 대학업무 중에서 입시관리 업무를 사용언어는 COBOL로 구현 하였다.

본 논문은 Ⅱ장에서 소프트웨어 재사용에 관한 기존의 기본기술과 방법을 고찰하여 재사용에 필요한 개념을 정립하고, 기존의 재사용 시스템이 안고 있는 미비점을 확인하고, Ⅲ장에서는 문제점의 해결 방안으로써 재사용을 지원

하는 응용 프로그램 재사용 시스템을 설계하고 구현하였다. Ⅳ장에서는 응용 프로그램 재사용 시스템을 실행시켜 결과를 확인하고, Ⅴ장에서는 본 연구에서 완성하지 못한 구현 부분 등의 향후 연구과제에 대해 언급 하였다.

Ⅱ. 소프트웨어 재사용의 기존연구

1. 재사용을 위한 기본이론

1.1 소프트웨어 재사용의 개념

소프트웨어의 재사용은 새로운 소프트웨어를 개발할 때 기존의 소프트웨어를 그대로 재사용하거나 상이한 일부분 만을 수정하여 다시 사용하는 것을 말한다. 컴퓨터의 응용분야가 확대되고 소프트웨어의 규모와 수준이 이전보다 복잡하고 어려워 졌고 따라서 프로그래머의 부담이 커지고 개발비용이 급증하는 문제를 해결하려고 활발히 연구하고 있는 분야가 소프트웨어 재사용을 통한 문제 해결 방법이다. Lanergan과 Poynton[1979]에 의하면 소스코드의 60%가 하나이상의 응용분야에서 반복 개발되고 있다 하였고, Kapur[1985]에 의하면 프로그램에 나타나는 70%의 함수가 유사함을 밝히고 있다. Biggerstaff[1984]에 의하면 응용프로그램의 원시코드중 40-60%가 재사용 가능한 공통된 부분으로 모든 상업용 프로그램 코드의 60%가 재사용 가능한 것으로 나타났으며, Jones[1984]에 의하면 원시코드중 15% 만이 특정 프로그램에 유일하고 85%는 그 기능이 공통적임을 밝히고 있다. Matsumoto[1986]에 의하면 일본 소프트웨어 공장의 프로그래머의 평균 생산능력은 프로그램을 신규로 개발 할 때는 500-800 SLOC/Month이지만 프로그램 코드를 재 사용할 때는 800-3200 SLOC/Month로 생산성이 뚜렷이 향상됨을 밝히고 있다. 이와 같이 각종의 소프트웨어는 40-60%정도가 같거나 유사 하다는 것을 알 수 있으며 유사한 소프트웨어를 약간씩 수정하여 재사용 한다면 프로그래머의 생산성을 저해하는 가장 큰 요인중의 하나인 소프트웨어의 신뢰성을 보장 받을 수 있고, 검증에 필요한 비용과 노력을 절약하고, 프로그래머의 부담이 경감 됨에 따라 개발기간의 단축은 바로 개발비용의 경감으로 직결되고 유지보수가 용이 하다는 등의 여러 가지 이점을 얻을 수 있다.[Ramamoorthy 1984; Horowitz 1984; Standish, 1984]

소프트웨어 재사용에 관한 개념은 1950년대의 서브루틴 라이브러리가 그 효시라 할수 있으며, 1961년에는 소프트웨어 부품목록이 나왔다. 1968년에는 맥로이가 소프트웨어 생산공장화, 즉 소프트웨어 부품공장을 제안 하였고 1976년도에는 소프트웨어 공장의 실현을 위한 작업들이 구체화 되었다. 1980년대에 들어와서 미 국방성의 Ada, 제록스의 Smalltalk-80등의 재사용 가능 소프트웨어를 개발하는 언어와 환경이 개발 되었고, 1982년 미 국방성의 자료에 따르면 소프트웨어의 개발 요구는 해마다 12%씩 증가 하는데 반하여 이를 해결할 전문인력은 년간 4% 정도로 증가 하므로 소프트웨어의 수요와 공급은 시간이 지날수록

불균형이 심화되리라 하였고, 소프트웨어의 유지보수량이 증가함에 따라 많은 인력과 경비가 소요되고 있으나 이를 해결할 어떤 방법론이나 도구는 부족한 실정이라 하였고, 1983년의 소프트웨어 재사용에 대한 ITT 워크샵 이후 각종 학술회의와 워크샵에서 소프트웨어 재사용 논문이 발표 되었다.

Standish[1984]는 'Code is not reusable' 이라 하였는바 그 당시에는 개인이 작성한 응용 프로그램을 수정없이 그대로 사용 하기에는 재사용의 적용 범위가 너무 적었고, 대부분의 경우 프로그램 자체가 너무 특정업무에 한정된 것 이라는 점에 기초한 개념이었다. 그러나 재사용 분야의 연구가 거듭됨에 따라 재사용의 의미도 수정없이 직접 재사용 하는 것은 물론이고 일부 수정후의 재사용을 포함하며 프로그래머가 자신이 작성한 코드를 본인이 직접 재사용하거나 단일 업무 내에서의 재사용, 서로 다른 업무간의 부품 재사용등을 포함한다. [Grasso, 1984; Horowitz, 1984; Gouthier, 1987]

1986년에는 미국방성 주관하의 STARS 워크샵에서 소프트웨어 재사용에 대한 지침서가 제안되고 1987-89년에는 ACM, COMPCON, COMPSAC, ISS, TOOLS등의 주요 학술회의에서 소프트웨어 재사용에 대한 많은 논문이 발표 되었다. [Anderson, 1989; Arnold, 1987; Chisio, 1987; Jameson, 1987; Traacz, 1987; Wald, 1987; Yoshinori, 1987]

1990년 부터 사용자의 편의성과 동시에 시스템 개발자의 개발 편의성과 생산성 향상에 중점을 둔 소프트웨어 재이용에 관한 연구가 과기처에 보고되고 재사용 시스템 도구인 CARS가 개발되어 재사용에 대한 관심이 고조되고 연구가 활발히 진행되고 있다.[이경환, 1989; 변상용, 1990; 중앙대학교, 1993]

1.2. 소프트웨어 재사용의 대상

소프트웨어의 재사용 대상은 소프트웨어 수명주기 전 단계에 걸쳐서 생성된 산물이어야 하지만 현재까지 완벽한 재사용 지원도구는 없고 수명주기의 일부분에 걸쳐서 연구되고 있는 바, Freeman[1987]의 연구에 의하면 소프트웨어 개발과정에서 재사용 가능한 정보는 환경적 지식, 외부적 영역지식, 기능적 구조, 논리적 구조, 코드조각이다.

코드조각의 재사용은 프로그래머의 산물인 원시코드를 재사용 하는 것으로 가장 먼저 발전된 방식으로 초기에는 라이브러리 함수의 재사용을 시발점으로 하여 현재는 기존 프로그램의 원시코드 부분을 재사용 하도록 지원하는 도구가 등장하였다. 논리적 구조는 시스템의 내부설계 부분으로 소프트웨어의 프로세스와 데이터 구조로 구성된 정보로 재사용 부품과 그들 사이의 관계를 정의한 것으로 대응된 자료를 변화시키면서 논리구조(프로그램과 자료구조)를 재사용 한다. 기능적 구조는 내부설계에 익숙지 않은 외부 사용자를 위한 모듈의 기능을 설명한 외부설계로 이 수준의 정보는 기능과 자료객체의 명세가 있고 재사용 정보에는 기능적 집합과 포괄적 시스템이 있다. 외부영

역지식은 입시관리나 도서관 업무관리등과 같은 응용 소프트웨어를 개발 한다고 가정할 때 이 들 입시나 도서관 관리에 관한 지식과 같이 대상영역에 대한 지식과 아울러 소프트웨어를 개발하는 기술과 절차에 관한 지식을 말한다. 환경적인 지식은 소프트웨어 신 기술에 대한 기술이전에 관한 지식과 소프트웨어를 실제 사용하는 방법에 대한 사용지식을 포함한다. [Jameson,1989]

소프트웨어 재사용이 가장 먼저 시도된 부품은 프로그램 원시코드의 일부 였으며, 그 다음이 하나의 기능을 수행하는 원시코드의 블록을 하나의 모듈로 보고 입출력과 수행기능만 알면 이를 블랙박스화 하여 모듈의 재사용은 가능하다고 보는 모듈의 재사용이 있고, 시스템에 내장된 서브루틴이나 표준화 된 함수들이 있었고, 프로그램 전체를 재사용하기도 하였다. 프로그램의 원시코드나 모듈을 재사용하는 실례를 보면 Raytheon회사는 3,200개의 COBOL 원시코드 모듈을 재사용하며, Harford보험회사는 20개의 서브루틴과 15개의 프로그램등 도합 35개의 원시코드 모듈을 재사용하며, AT&T는 1,000개의 C언어 부품을 재사용하며, GTE사는 220개의 재사용 가능부품, 960,000라인에 이르는 코볼, C, 어셈블리 부품을 재사용 하고 TI사는 30개의 엔티티 정의와 40개의 트랜잭션, 15개의 일괄처리 절차로 구성된 IEF 설계모델을 제공하고 있다. 사용자가 제시한 요구사항은 사용자가 필요로 하는 기능과 제약사항, 사용자와 시스템간의 입출력 메세지 등을 포함하며 개발 담당자는 컴퓨터 시스템에서 수행가능한 기능과 활동이 무엇인가를 찾아서 개발기능으로 재정의하는바 이 시스템의 외부기능은 시스템이 수행할 기능을 외부로 나타내는 행위가 되고 운영체제와 같은 내부기능도 정리해 둔다. 이 개발기능이 블록으로 변환되고 블럭이 기본적인 소프트웨어 개발 단위이다.[Zave, 1984]

원시코드 재사용의 약점을 보완하기 위해서 연구되고 있는 분야 중에는 설계정보 재사용에 관한 연구가 있으며 설계정보의 재사용은 프로그램 코딩의 앞 단계인 요구분석과 설계단계에서 생성된 산출물을 재사용하자는 의도에서 연구되고 있으며 설계단계에서 산출되는 모듈에 대한 기능명세서, 입출력 자료의 명세등을 재사용하면 프로그램의 원시코드 재사용 보다는 자료형이나 프로그래밍 언어와 무관하게 사용할 수 있는 장점을 가진다. 설계정보를 재사용하려면 예비 설계단계에서 생성되는 정보는 주석을 달은 호출목으로 표현하며 상세 설계단계에서는 설계파일과 그룹파일로 표현하고 이들간의 상호 변환도구를 제안하였다. 호출목이란 설계모듈간의 호출 계층관계를 처리하고 매개변수와 자료형을 유지하고, 설계파일은 설계모듈의 알고리즘을 포함하고 컴파일 가능한 모듈이며, 이 컴파일 가능 모듈로 부터 역으로 설계파일로 변환하는 도구를 제안 했다. 설계단계에서 생성되는 산출물들은 모듈에 대한 기능명세와 입출력 자료에 대한 명세, 모듈간의 관련성 인데 설계정보를 재사용 하려면 예비 설계단계의 산출물은 주석달린 호출목으로, 상세 설계단계의 산출물은 설계파일과 그룹파일로

표현하고 이들 간의 상호변환 도구들을 제안하고 있다.[Charette, 1986; Wald, 1987; Jameson, 1989]

그러나 이 도구들은 텍스트에 기반을 두기 때문에 사용자의 편의성, 재사용에 필요한 인터페이스, 설계정보의 분류나 검색방안, 기존 방법론과의 연계면에서 한계가 있다. 소프트웨어 개발시 재사용 할 수 있는 대상은 소프트웨어 수명주기 각 단계에서 생산된 산출물중 재사용이 가능한 부분들을 망라 한다. 소프트웨어 수명주기별로 재사용의 대상과 필요성을 살펴보면 다음과 같다.

(1) 소프트웨어 분석 단계 : 컴퓨터의 응용분야가 점점 광범위해지고 개발할 소프트웨어의 규모와 수준이 크고 높기 때문에 새로운 소프트웨어의 기능에 대한 사용자 측의 요구사항에 대한 명세서를 토대로 이를 분석하여 요구사항을 해결하기 위한 소프트웨어의 사양과 필요한 시스템의 성능과 제한사항들을 정의한다. 분석단계의 중요한 산출물은 각각의 기능에 대한 명세서이다. 어떤 요구는 소프트웨어의 구현이 불가능 할 수도 있고 어떤 경우는 다음 단계인 설계 과정에서 변경될 수도 있다. 사용자의 요구사항에 대한 분석은 나머지의 생명주기 전반에 지대한 영향을 미치게 되므로 요구사항의 정확한 정의와 발견된 문제점이 무엇인가를 정확히 분석하고 이에 대한 해결책을 세워야 한다. 분석후 정리 해야할 서류들은 자료처리의 흐름을 표현한 자료흐름도와 자료사전을 작성하는데 대한 분석도구들이 필요하다.

(2) 소프트웨어 설계 단계 : 분석단계에서 생성한 자료 흐름도에 근거하여 프로그램의 설계를 각 요소의 계층적 구조로 표현하여 기능 시스템의 구조도를 표현하고 이에 따른 프로세스 기능을 설명하는 프로세스 명세서에 관한 구조적 도표를 작성한다. 데이터 흐름도로 부터 구조도를 작성하기 위해서는 변환분석과 처리분석, 변환분석과 처리분석의 조합의 단계를 거쳐 이를 평가하고 구현단계에서 설계도가 유용하게 쓰이도록 설계 정리작업을 한다. 이를 지원 하려면 자료흐름도에서 구조도를 작성하는 것과 구조도의 각 모듈과 관련된 모듈명세 작성 지원 도구등이 필요하다.

(3) 소프트웨어 코딩 단계 : 자료흐름도를 구조적 설계에 바로 적용 하기는 곤란하여 트리구조로 시스템을 표현하고 각 모듈의 구체적 내용은 구조화 영문이나 가상코드로 설명한 명세서를 작성할 때 일상적인 용어를 사용하면 그것에서 수반되는 여러 가지 모호성이 있으므로 이를 피하기 위해서 구조화 영문을 사용한다. 구조적 영문도 다른 기법과 같이 기본적인 네가지의 제어 구조를 가지며 간단한 연속구조, 주어진 조건에 따라 반응하는 조건구조, 여러 가지 경우중 한가지를 선택하는 선택구조, 종료할 조건을 명시한 문장의 반복구조로 구조화 영문 사용규칙에 따라 가상코드로 작성한다. 코딩단계에서는 구조도로 부터 변환하거나 개별적으로 작성한 원시코드가 생성되며 구조도로 부터 변환시에는 가상코드를 프로그램으로 변환하는 도구와 각종 언어로 원시코드를 작성 할 때는 각 언어의 문법을 기반으로 하여 프로그램을 편집하는 프로그램 편집기, 디버깅

도구, 컴파일러등이 필요하다.

(4) 소프트웨어 검증 단계 : 컴파일이 완료된 소프트웨어 모듈은 분석,설계단계에서 설정한 기능을 올바르게 수행하는가를 단위모듈과 통합 시스템으로 검증을 하고 테스트 자료와 그 결과를 정리할 필요가 있다. 구조적 설계와 하향식 기법의 차이, 잭슨기법과 와니어-오어 기법의 비교를 통하여 보다 나은 기법을 지원하는 도구가 필요하다. 테스트를 위한 자료를 생성하고 그 자료에 의한 처리 결과를 산출하는 도구들이 필요하다.

(5) 소프트웨어 유지보수 단계 : 처리 결과가 산출되면 사용자가 원하는 그대로의 기능을 만족 하는지 아닌지가 증명되고 기능의 추가 사항이나 불만족 사항에 대한 유지보수 작업이 필요하다. Lientz[1987]의 연구에 의하면 총 유지보수 비용의 42%는 사용자의 요구가 변경되어 보수작업이 필요하고 17% 정도는 데이터 형이 변경되었고, 그리고 12% 정도는 응급처치에 의한 유지보수 작업이라고 밝히고 있다. 소프트웨어의 유지보수를 최소화 하려면 사용자의 요구를 보다 정확하게 확인하고 사용자와 개발자간에 의사소통의 불분명함으로 빚어지는 요구정의의 모호성을 줄이는 방법과 도구가 필요하다.

1.3. 소프트웨어 재사용 기술

소프트웨어 재사용 기술에는 재사용 되는 구성요소의 성질에 따라 두가지로 구분 할 수 있는데 그 하나는 상향식으로써, 모듈로 부터 시작해서 불록을 만들어 가는 빌딩 불록 형식으로 합성을 해 가는 합성기술과, 하향식이라고도 하는 구조로 부터 생성해 내는 방식 즉, 기능으로 부터 응용 업무에 공통으로 쓸 수 있는 구조의 뼈대를 만들고 이 뼈대에 맞는 새로운 프로그램을 생성하는 생성기술로 대별 할 수가 있다. 합성기술은 기존의 재사용 라이브러리로 부터 검색된 부품을 사용하여 새로운 소프트웨어를 만들어 내는 기술이고 생성기술은 응용 소프트웨어 생성기 같은 도구를 통해서 재사용 부품들이 만들어 지며 부품의 검색, 과정등은 대부분 자동화하는 기술을 말한다.[Richter, 1987] 이외에도 재사용과 관련된 기술에는 알맞은 재사용 부품을 찾아 내는데 관한 검색분야에 ESLREX가있고 기존의 소프트웨어 재사용 부품을 쓰고자 할 때 사용자에게 해당 모듈의 이해를 돕는 'C Information abstractor'와 같은 시스템에 관한 기술이 있다. [Chen, 1986 ; Biggerstaff, 1987]또한 객체지향 설계나, 영역분석, 변수화와 상속, 모듈화등의 기술이 정립 되고 재사용 소프트웨어 라이브러리와 카타로그를 생성하고 관리하는 종합적인 지원환경 구축에 관한 기술과 재사용 부품의 개발과 이의 이용에 관한 지침서가 제시되고 있다.[UCLA, 1988]

합성기술(Composition technologies)에서 재사용하는 부품은 잘 정의된 규칙에 의해 구성된 원자로 된 빌딩불록으로 그들을 재사용할 때 각각의 식별자가 그대로 존속되기 때문에 원자라 하며 잘 정의된 경우 수정없이 재사용이 가능하고 정적이다. 프로그램과 코드조각이나 서브루틴과 함수, Smalltalk의 객체등의

재사용 부품은 새로운 프로그램을 생성할 때 기본이 되는 빌딩블록이며 합성원칙을 적용하여 합성하게 된다. 합성기술을 적용한 대표적인 경우가 유닉스의 파이프 메커니즘과 Smalltalk의 메시지 전송 및 상속 메커니즘이다. 이 기술은 중점을 두는 분야에 따라 2가지로 대별 할 수 있는데 하나는 부품 자체의 개발과 축적부분, 즉 응용 가능한 부품의 라이브러리를 강조하고, 다른 또 하나는 재사용 부품을 구성하는 일반적인 구조나 합성규칙을 강조하는 분야이다.

(1) 응용 부품 라이브러리(Application component labraries) : Lanergan[1984]에 의하면 부품의 규격과 합성의 정형화 보다는 유용한 응용 부품의 라이브러리를 만드는 것에 중점을 두었으며 부품 규격의 정형화는 COBOL을 직접 사용하고 합성 규칙은 프로그래밍 언어가 결정한다. Yoshinori[1987]에 의하면 부품의 규격 메커니즘과 이 들의 검색 용이성에 중점을 두고 요구사항에 관한 부품, 설계 부품, 프로그램 부품들에 대한 라이브러리를 강조 하였다.

(2) 구조 및 합성원리(Organization and Composition principles) : 단순한 프로그램을 보다 복잡한 프로그램구조로 연결 하려는 유닉스의 파이프 메커니즘처럼 부품들이 필요로 하는 목적 프로그램으로 결합되는 합성원리와 구조에 중점을 둔다. 초기의 유닉스 개발 환경은 어느 하나의 프로그램에 유용한 어드레스 영역이 매우 한정적이라 큰 연산을 수행하기가 불편 하였으므로 필터를 이용한 구조화된 프로그램으로 발전 했고 파이프 연결자를 이용하여 명령어 수준에서 그들을 합성 하였다.

생성 기술(Generation Technologies)에서 재사용 되는 부품은 여러 분야에 적응 시킬 수 있고 순응성 있는 동적인 것이고 재사용되는 패턴은 코드 패턴과 변환규칙 패턴의 두가지가 있다. 두가지 패턴 모두다 재사용 부품의 효과는 합성기술의 빌딩 블록 효과 보다 광범위 하고 보다 확산되는 경향이 있어서 각 부품의 식별과 재사용의 원리를 특징 짓는 간단한 정의 방법이 없다. 이 그룹은 중점 분야에 따라 3가지로 세분 할 수 있다.

(1) 언어 기반 시스템(Language-based system) : 언어에 기반을 둔 시스템은 사용자가 코드 생성기의 구조나 원리를 이해하기 보다는 언어의 규칙을 이해하고 이를 기술하는 표현법을 강조한다. 사용되는 언어는 VHLL이라는 범용언어나 문제지향 언어에 이르기 까지 다양하다. VHLL언어는 말 그대로 어떤 특정영역에 국한되지 않는 넓은 분야에 적용 가능하고 문제 지향언어인 POL은 어떤 특정 문제에 한정된다.

(2) 응용 생성기(Application generators) : 응용 생성기는 제한된 형태의 문자나 사용자에 의해 상호작용적으로 제공된 입력을 받아 생성기내에서 이를 코드화 한다. 스크린 포맷터 처럼 사용자가 특별한 키나 특별한 코드로 입력한 스크린상의 필드에 대한 메타정보를 받아 이에 상응하는 스크린 구조를 실시간으로 만들어 주는 화면 구조를 기술하는데 만 전용으로 쓰는 응용 생성기와 개인용 컴퓨터의 파

일관리 시스템처럼 언어로 간주 하기가 힘든 간결한 필드를 기술하면 자체내에서 이를 코드화 하는 생성기도 있다.

(3) 변환 시스템 (Transformation systems) : 시스템 개발자가 이해하기 쉽도록 목적하는 시스템을 규격화함에 있어서 사용자는 간단한 규격을 제시하더라도 이를 효율적으로 실행가능한 형태로 재작성 해 주는 것이 변환 시스템이다. 변한 시스템은 변환을 수행하는 하나의 단순한 도구이고 이에 적용할 규칙은 데이터로 존재하다가 규칙이 제시되면 이에 맞는 하나를 찾으면 관련된 규칙을 적용하는 과정을 종료 조건이 나올 때 까지 반복하여 간단한 요구로 부터 완전한 프로그램으로 변환하는 시스템이다.

사용자가 소프트웨어 부품을 효율적으로 재사용 할 수 있도록 지원하자면 위에서 언급한 모듈을 이용한 합성기술과 구조를 이용한 생성기술을 구현하는 재사용 지원 시스템이 필요하며 이들을 두가지로 구분하면 다음과 같다.

(1) 모듈 이용 시스템 : 재사용 부품을 이용하여 효율적인 프로그램을 만들어 내려면 우선 갖추어야 할 것이 객체, 코드와 서브루틴, 프로그램과 기능등 고품질의 재사용 부품을 확보해야 하고 부품이 확보되면 이들을 쉽게 찾아서 재사용 하게 체계적으로 분류·저장해야 하고, 저장된 부품을 찾고 수정할 시스템이 있어야 한다. 새로운 프로그램의 합성시 먼저 요구를 분석하여 소프트웨어 구조를 설계하면 설계한 모듈과 일치하는 부품을 소프트웨어 라이브러리안에서 검색하여 찾아낸다. 만약 일치하는 부품이 없으면 비슷한 기능의 부품을 찾아내어 이를 수정하고, 비슷한 부품도 없다면 새로운 부품을 생산한다. 부품이 확정되면 이들을 결합하여 요구에 맞는 새로운 소프트웨어를 생성한다. 이러한 기능을 수행하는 시스템을 구성하려면 재사용 가능 라이브러리와 사용자 질의 서브 시스템과 소프트웨어 부품의 평가 서브 시스템등이 있어야 하고 이 시스템이 재사용을 지원하는 알고리즘은 다음과 같다.

Given a set of specs
Begin
search library if identical match then terminate else
collect similar components for each component compute degree of match end
rank and sellect best modify component endif End

이 기술은 정보은닉과 객체 지향 프로그래밍 개념을 도입한 일반적인 프로그래밍 방법과 기능이 유사한 프로그램들을 모아놓고 이들의 공통성을 분석하고 사용자의 매개변수와 합성규칙을 정의하고 이 들을 결합하는 매개 변수화한 프로그램이 있다. 이 방법은 사용자의 매개변수와 매개변수 값들을 합성법칙으로 삼고 이 법칙에 따라 결합된 프로그램을 합성해 나갈 선행 처리기가 있다. 이러한 방법을 쓰면 코드 단계의 부품을 재사용 하는 것이 활성화 되고 규모는 작지만 효율이 더 좋은 프로그램을 생성 할 수 있다. 그러기 위해서는 사용자의 지침이 될 선행 처리기와 저장된 부품에 대한 문서화가 필수적이다. 또 한가지 방법은 프로그램

의 제어구조를 나타내는 모형틀을 만들어 놓고 사용자는 이 틀을 이용하여 필요한 프로그램 부분의 기능성에 대한 정보를 보강하여 프로그램의 일반적인 부분을 인식하고 최종적으로 공란을 메워 새 프로그램을 생성한다. 이 방법은 사용자의 요구에 쉽게 적응 가능하고 재사용 빈도에 비례하여 생산성이 증가하며 모형틀과 테이블을 이용한 즉석 사용의 대체 기법이 재사용 가능하다는 장점이 있는 반면, 나중에 개발한 골격 프로그램을 즉각 재사용함에 따르는 신뢰성 문제, 즉 수정·보강 작업이 곧바로 뒤따르는 단점이 있다.

(2) 구조 이용 방법 : 응용분야에서 관련된 유용한 기능을 종합하여 뼈대라는 재사용 구조를 만들어 두고 사용자는 이를 이용하여 필요한 부품을 생성하는 법이다. 여기서의 구조란 설계의 기술 또는 과정이나 구성의 형태 또는 같은 타입 객체들의 구조를 말하며 소프트웨어 구조란 소프트웨어 분석 설계자가 사용한 요소들과 부품을 합성할 때 사용된 규칙들의 집합체를 말하는바 구조란 기능으로 부터 형식으로의 전환을 의미한다.

2. 재사용 관련분야 연구동향

2.1. 소프트웨어 개발 방법과 재사용 기술

소프트웨어의 개발 방법에 대한 발전과정을 보면 단일 업무의 간단한 프로그램을 개별적으로 작성 할 때는 별반 문제가 없었으나 큰 프로젝트의 경우 개발에 참여한 여러 명의 프로그래머가 여러 방법으로 작성한 프로그램을 하나로 통합 한다는 것은 매우 힘든 일이었으며, 소프트웨어의 생산성을 향상 시킨다는 측면에서 서브루틴이나 함수를 만들어 놓고 이를 공용하는 모듈화 프로그래밍 방법이 등장 하였다. 또 한가지 방법은 하나의 큰 프로그램을 여러 개의 기능별로 분해하여 각 프로그램을 구조적인 체계로 만든 구조적 프로그래밍 방법에 의해 개발시나 유지보수시에 효과적으로 대처 하였으나 적용영역이 너무 협소하였다.

Fisher[1989]에 의하여 컴퓨터가 기능적 분해과정을 담당하고 시스템의 분석과 설계, 구현과 테스트 단계까지 프로그램과 모든 문서를 포함한 소프트웨어 전체를 자동으로 생산 하자는 의도에서 CASE가 등장 하였고 이에 대한 연구가 현재 활발히 진행되고 있으나 실제로 소프트웨어 개발 전과정을 지원할 자동 생성기의 구현이 어렵고 자동화 이전에 소프트웨어에 관한 모든 문서나 방식의 표준화가 선행 되어야 가능하다.[Gane, 1990] 경험이 적은 프로그래머를 위하여 응용 프로그램을 자동 생성하고 각종 양식이나 메뉴, 각종의 리포트를 자동으로 만들어 주는 제 4 세대언어가 있으나 해결할 문제점을 정확히 분석하여 시스템의 기능에 대한 요구 정의를 4세대가 정의한 형식과 일치 시키기란 쉽지 않다. 위와 같은 여러 가지 개발 방법과 병행하여 소프트웨어 재사용 기술은 일반 함수의 재사용이나 구성요소의 재사용, 프로그램 변환에 의한 재사용, 파일 전송에 의한 재사용등으로 부분적으로 사용 되어왔고 이 방법을 객체지향 파라다임에 의해 접근하면 생산성 향상에 크게 기여하리라 하였다.[이경

환, 1993]

(1) 소프트웨어 재사용 가능성과 도구에 관한 연구 : 소프트웨어 재사용에 관한 연구의 궁극적인 목적은 고품질 소프트웨어의 생산성 향상이다. 생산성의 향상이란 소프트웨어의 개발기간의 단축과 신 제품의 개발비용이나 이의 유지보수 비용의 절감을 말한다. 소프트웨어 재사용에 관한 연구에서 가장 먼저 접근해야할 부분이 재사용 가능성에 대한 연구이다. 소프트웨어 수명주기 중에서 잠재된 생산성 향상 가능성을 개발비용 중심으로 살펴보면 소프트웨어 요구로 부터 시스템 테스트에 이르는 전 과정 중에서 40% 정도의 생산성 향상이 가능하고 그 중에서도 코딩 단계에서의 잠재적 생산 가능성은 75%로 최고의 생산성 향상을 도모 할 수 있는 단계라고 하였다. 소프트웨어의 유지·보수비용은 개발비의 3배 이상이므로 소프트웨어 재사용을 이용한 생산성 향상은 개발중은 물론 개발후의 유지.보수단계에서도 잠재적 생산성 향상이 된다.

소프트웨어의 위기를 해결하려면 기본적으로 프로그래밍 방법을 개선해야 하고 예상되는 한가지 해결책으로 '재사용 가능 소프트웨어의 개념'에 생산성을 기대해야 한다고 하였다. 소프트웨어의 원시코드는 중복성이 많기 때문에 기존의 원시코드들을 분석하여 고품질의 재사용 코드 라이브러리를 구축하고 이를 활용해야 하며 장래에는 소프트웨어 재사용 코드 라이브러리를 구축하지 않는 소프트웨어 회사들은 그들 사업에 중대한 위기를 맞을 것이라 하였다. 대형의 응용업무인 경우에 현재는 36개월 정도 소요되지만 90년대에는 재사용 라이브러리의 이용으로 개발기간이 6개월 정도로 단축되리라 하였다. [Horowitz, 1984: Munson, 1984: Ninamary, 1986: Chisio, 1987]

Tracz[1987]에 의하면 재사용을 위한 기준에 따라 개발된 모듈은 실행 횟수가 많아지며 따라서 에러율은 낮아지고 신뢰성은 높아진다고 하였으며 실행횟수가 많은 프로그램 일수록 더 많은 검증을 거친 코드이며, 별로 사용하지 않은 프로그램 코드는 비교적 검증횟수가 적을 뿐더러 신뢰성도 그 만큼 낮다. 소프트웨어가 재사용 되기 위해 표준화된 설계와 코딩, 문서화등이 잘 되고 체계적으로 구축 된 경우 이를 잘 활용하면 소프트웨어의 생산성과 신뢰성은 매우 높아진다.[Mitchel, 1986] 일반적으로 재사용 부품의 주요 대상은 데이터나 프로그램 모듈, 이들과 관련된 문서화, 테스트 케이스등이며 지금과 같은 추세로 재사용에 관한 연구가 진전 된다면 90년대에는 상업용 프로그램 중 전체 코드의 50%는 재사용 논리구조와 재사용 코드가 차지할 것인바 그러기 위해서는 이미 존재하는 재사용 가능 코드를 찾고 이들을 분류 하는 등의 지원을 담당할 도구의 개발이 뒤따라야 한다.[Jones, 1984]

소프트웨어 재사용 시스템으로 구현 가능한 시스템 모형은 합성기술과 생성기술의 두가지 모형이 있으며 합성기술을 적용한 모형은 그림 1과 같다.

재사용 소프트웨어가 갖출 내용은 재사용 부품의 확인을 위한 검색 메커니즘과 정의된 기능은 사용자가 쉽게 이해할 수 있도록 표현하

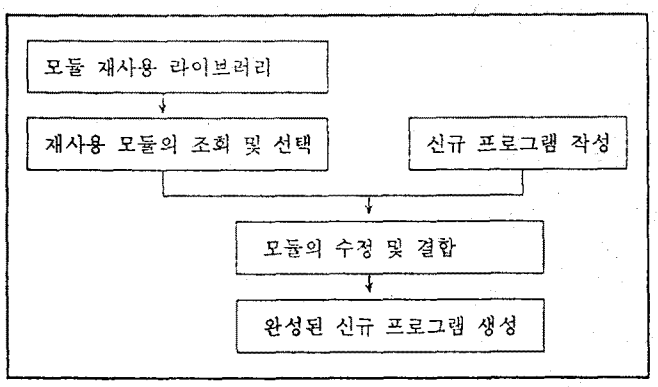


그림 1. 모듈 재사용 시스템 모형

는 방법과 부품의 저장시 적용범위에 따라 적절히 분류저장하는 방법, 그리고 부품을 구현한 언어에 대하여 언급해야 한다.[Munson, 1984]

재사용 기법을 소프트웨어 개발에 적용하려면 이를 위한 재사용 시스템이 갖추어 져야 하는데 재사용 시스템이 다루어야할 기본적인 항목들은 재사용 부품의 기능과 사용시기, 재사용 부품의 존재여부 인식, 재사용 부품에 접근하는 방법, 해당 부품의 변경이나 확인된 부품의 합성방법등이 갖추어 져야 한다. [Biggerstaff, 1987] 이 시스템에서 사용할 재사용 라이브러리를 구축 할 때는 일반적으로 자주 사용되는 기능루틴과 포괄적이며 유일한 모듈을 구성하고 고급언어 모듈과 공통루틴들을 포함한 다음 이들을 접근하여 재사용 할 수 있는 메뉴방식의 생성기등이 필요하다. [Maginis, 1986]

소프트웨어 개발시에 코드나 모듈을 재사용하는 기법을 사용하기 위해서는 사용자의 요구를 시스템의 기능으로 분석하고 기능을 수행할 소프트웨어의 구조를 설계한 다음에 아래와 같은 3단계를 거치는 것이 일반적인 방법이다.

첫째는 검색 단계로 설계한 모듈과 일치하는 코드 부품을 저장된 부품 라이브러리에서 찾아야 하고 꼭 맞은 부품이면 그대로 사용하고 유사한 부품이면 일부 수정후 사용하며 유사한 부품도 없을 때는 새로운 부품을 개발해야 한다. 둘째, 수정 단계로 유사한 부품을 찾은 경우에는 이를 소프트웨어 구조에 맞게 수정해야 한다. 마지막 단계에서는 새로 개발한 코드와 재사용 부품을 조합하여 완전한 소프트웨어로 구축한다.

원시코드 재사용의 약점을 보완하기 위해서 연구되고 있는 분야 중에는 그림 2와 같은 시스템 구조의 설계정보 재사용이 있다. 설계정보 재사용 시스템은 시스템 분석과 설계단계에서 생성되는 부품을 재사용 하자는 것으로 사용자는 그래픽 에디터와 전후진 검색기에 의해 설계정보 라이브러리에 접근하여 구조도 라이브러리와 수정된 모듈 명세서 라이브러리를 생성할 수 있고, 텍스트 에디터를 통하여 수정된 모듈 명세서 라이브러리와 원시코드 라이브러리에 접근하여 재사용 가능 원시코드를 생성할

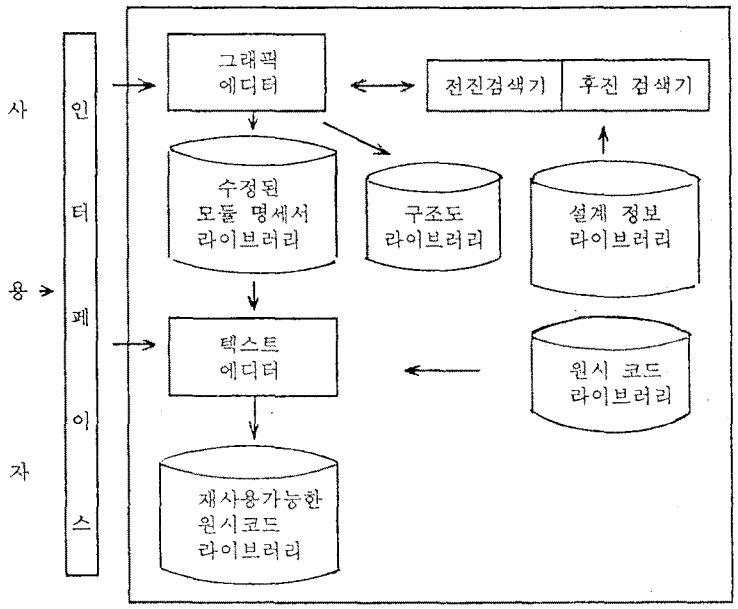


그림 2. 모듈 재사용 시스템 모형

수 있다.

신규로 소프트웨어 재사용 부품을 만들 때는 기존의 고품질 소프트웨어를 이용하여 만드는 방법과 기존의 소프트웨어가 없을 경우에는 신규로 생산해야 하는데 이때는 신규작성의 준칙과 같은 표준화를 위한 어떤 기준이나 규칙에 의하여 작성이 되야 한다. 기존의 부품을 재사용하여 또 다른 부품을 생산할 때는 환경요소가 중요한 기준이 되며 동질의 환경이라면 기존의 재사용 소프트웨어 중에서 적당한 것을 검색해 사용하고, 동질의 것이 없는 이질환경이면 유사한 부품을 수정하여 환경을 일치 시켜야 한다. 그러기 위해서는 적합한 부품을 검색하여 찾아 내거나 비슷한 부품을 수정 할 수 있는 어떤 도구가 있어야 한다.

신규작성을 위한 기본적인 기준 혹은 고려사항에는 다음과 같은 것이 있다. 먼저 하나의 부품은 단 한가지 기능만을 수행하게 한정한다. 둘째로 부품은 환경의 변화에 독립적 이어야한다. 셋째로 부품의 명확한 인터페이스에 대한 언급이 필요하다. 예를 들면 부품간의 인터페이스로 유닉스의 파이프와 같은 메시지 열로 한다 던지 하는 그러한 기준이 미리 정해져야 한다는 점이다.

(2) 범용 언어에 의한 소프트웨어 재사용 연구 : 일반 상업용이나 사무용 프로그램들은 업무의 특성상 코드의 중복이 거의 없는 것처럼 인식하고 있으나 Raytheon Missile System Division[1979]의 연구에 의하면 5,300 여개의 COBOL 프로그램을 대상으로 분석한 결과 갱신 프로그램은 24%, 편집 프로그램은 30%, 보고서 작성 프로그램은 46%가 반

복적인 논리구조 또는 기능이 있음을 확인하였다. 따라서 갱신과 선택, 보고서 작성등의 논리구조를 중심으로 COBOL 언어에서 편집, 갱신, 보고서 작성의 3가지 형태의 구조를 제안하였다. 일반적인 기능 모듈은 기능 루틴을 포함하여 파일 설명, 레코드 설명등의 기능, 입/출력 루틴기능이 있고, DB의 인터페이스 루틴과 입/출력 루틴기능, 인수 탐색, 처리 절차 부분의 호출등의 재사용 가능 기능모듈을 만든후 이를 이용하여 50%의 생산성 향상을 얻었다. [Lanergan, 1984; Frakes, 1987]

Kapur[1985]는 1,600 여개의 프로그램을 대상으로 조사 분석한 결과, 보고서 작성 프로그램은 40%, 갱신 프로그램은 27%, 자료 선택이나 편집 프로그램은 21%, 자료 추출은 12%비율을 얻었으며 이들을 토대로 일반적인 논리구조의 모듈을 작성하고 이를 재사용 했으므로 50−80%의 생산성 향상을 기대 할 수 있다고 하였다. 범용언어를 사용할 경우에 재사용 가능한 일반논리를 부분별로 보면, 편집과 데이터의 유효성 검증 논리가 있고, 순차 파일의 갱신과 VSAM 파일의 갱신, 그리고 DBMS 갱신등의 갱신 논리가 있다. 테이블 조작 논리로서 테이블 적재와 테이블 분류, 그리고 테이블 탐색 논리가 있다. 보고서 작성 논리에는 각주 달기나 페이지 달기, 그리고 인쇄용지의 여백조정이 있고 출력정보 제어선과 제어차단 논리를 제시하였고 실시간 처리로는 자료 조회, 자료 수집, 자료의 유효성 검증, 자료 보고서 작성으로 분류 제안 하였다.[Arthur, 1985]

이때 까지만 해도 재사용의 개념은 대부분 코드나 서브루틴 라이브러리의 재사용 형태이거나 각 루틴의 정의를 간단히 한다거나 인터페이스의 간편화에 중점이 주어졌다.

(3) 객체 지향 방법에 의한 소프트웨어 재사용에 관한 연구 : 객체의 개념은 1960년대에 개발된 SIMULA언어에서 도입 되었던 바, SIMULA는 모듈화에 중점을 두고 이제까지 모듈이 프로시져 위주로 작성하던 것을 바꾸어서 실질적인 객체를 기반으로 모듈을 작성하면 하나의 객체는 데이터와 대응하는 하나의 변수와 이 변수와 관련된 프로시져인 수행방법의 집합체가 된다. 객체지향 프로그래밍에서 중요한 개념의 하나인 객체는 절차지향 언어에서의 자료와 알고리즘의 통합체라 할 수 있는데 그 이유는 객체 자체가 처리할 자료와 그 자료에 관한 연산처리가 합해진 형태이다. 변수의 자료형을 추상적으로 표현해 놓고 프로그래머가 사용할 때 변수의 값에 따라서 그 타입을 정 해줄수 있는 고정된 자료형을 추상자료형이라 한다.

메시지는 활동하는 객체라 하며 수임자와 수임자의 실행 방법인 메소드, 이 메소드가 해당 기능을 수행하게 하는 자료요소들의 모임인 파라미터라는 세가지 요소로 구성되며 이때 메시지를 만들어 보내는 쪽을 발송자라 하고 수신하는 객체를 수임자라 한다. 클래스는 수행 방법과 변수를 정의하여 구체적인 객체타입으로 정해진 것에 부여하는 명칭인데 이 클래스에 실제적인 어떤 값을 부여하고 부여된 값과 대응된 클래스를 포함한다. 객체지향 언어(OOL)는 주로 Smalltalk와 C++가 널리 쓰

이며 이외에도 데이터 보호를 목적으로 하여 클래스의 개념을 최초로 도입한 SIMULA 67이 있고, 그 뒤에 Objective Pascal과 C++가 개발되었고 1980년대로 접어들어서 독자적으로 객체지향 언어의 설계기능을 가진 LISP와 Smalltalk를 결합하여 만든 FLAVORS, 그리고 LOOPS가 있다. 그 뒤를 이어서 C언어에 Smalltalk의 객체지향 설계기능을 접목시킨 objective C가 있고, Ada, Eiffel, CHILL등 여러 가지가 있다. 객체지향 설계(OOD)란 기존의 기능지향 설계(Function-oriented design)가 서브 프로그램의 기능적인 입장에서 연산처리를 위한 알고리즘의 추상화에 치중하던 것을 클래스의 계층 구조에 의해서 데이터와 프로그램을 통합하며 이들 클래스가 객체생성의 기준이고 객체가 공유할 프로그램의 저장장소도 클래스의 계층구조에 따라 설계하는 기법, 즉 소프트웨어의 코딩이나 설계등을 객체의 개념에 중점을 두고 실행하는 설계방법이다. 객체지향 설계에서는 객체들이 추상화와 정보은닉의 특성을 만족하도록 객체나 객체의 클래스로 시스템을 분해한다.

객체지향 설계의 중요한 단계를 보면 다음의 5단계로 구분된다. 첫째, 순항제어 시스템에서의 액셀레이터, 엔진, 속도와 같이 주로 명사형태의 객체에 대한 정의를 하고 이들의 특성을 결정한다. 다중 윈도우상에서 사용자 인터페이스를 설계한다고 할 때 도우미 윈도우, 메시지 윈도우, 명령 윈도우등을 각각 정의 하듯이 중복된 기능들도 각각 다른 객체로 따로 정의한다. 둘째는 객체들의 활동을 특성화하는 객체들이 제공하거나 객체에 필요한 연산처리 기능들을 정의한다. 다중 윈도우상에서의 Open, Close, Move, Size등이 윈도우상에서 정의해야 할 객체들의 연산처리 기능이다. 셋째로 객체를 하나의 독립된 객체로 사용할 수 있게 그 관계를 설명하여 객체에서 무엇을 할 수 있는가를 결정한다. 넷째로 객체를 사용자들이 쉽게 접근하도록 인터페이스를 만들고 각 객체의 내외적인 관계와 모듈양식을 정의한다.

마지막으로 객체에 대한 표현방법을 택하여 앞단계를 적용하여 실제로 객체를 실행한다. 객체지향 설계는 소프트웨어 수명주기중에서 설계와 코딩단계를 포함해서 적용하고 모듈을 분해하거나 이들을 결합시켜 새로운 클래스들을 생성한다.[Pinson, 1988 ; Rumbaugh, 1991]

2.2. 소프트웨어 재사용 도구 활용방법

(1) 응용 생성기(Application Generator)의 이용 : 응용 생성기는 NOMAD, RAMIS, DBASE, FOCUS처럼 모니터에서 대화방식의 인터페이스를 이용하여 데이터 베이스 질의어와 조작어를 사용하여 처리결과의 산출물이나 보고서등을 작성한다. 응용 프로그램을 개발하고자 할 때 그 프로그램을 최종적으로 사용할 사용자와 시스템 개발 담당자인 시스템 분석가나 프로그래머와의 사이에서 발생하기 쉬운 요구정의의 명세화 단계에서의 오류를 없애고 시스템 분석가나 프로그래머의 개입이 없이 사용자가 직접 필요로 하는 응용 프로그램을 생성하게 하자는 의도에서 주어진 요구정의에 따라

응용 시스템을 자동으로 생성 하는 것이 응용 생성기이다. 응용 생성기는 사용자 자신이 직접 제4세대 언어등을 통하여 요구정의를 하면 정보 저장소에 기존에 있던 프로그램을 이용하여 결과와 문서화된 서류들을 출력한다. [Woodfield, 1987]

(2) 분석 시스템(PSL/PSA,SREM)의 이용 : 개발하고자 하는 신 시스템을 프로세스의 집합으로 보고 각각의 프로세스와 데이터의 관계를 자료흐름도로 표현하고 자료 흐름도상의 각 요소에 대한 정의는 자료사전에 수집 저장한 후에 DFD와 DD로 구성된 시스템이 완성되면 기초적인 프로세스는 최소 규격으로 명세된다.

시스템의 요구정의 과정에서 부터 사용자의 동참을 유도하고 사용자가 요구한 내용을 분석한 후 결과를 시스템 모델의 형태로 작성하므로 사용자나 개발자 모두에게 공통적인 이해를 얻고 기능적 분해방법과 하향식 분석방법을 이용한 구조적 분석기법으로 시스템 요구를 정의하는 시스템이다. 구조적 분석기법은 도형적 모델을 이용하므로 사용자가 쉽게 시스템을 이해하게되고 분석가와 사용자간의 의사소통을 원활하게 하여 분석과정을 돕고 효율적인 시스템의 설계시에도 유용하다.[Hall, 1987; Kaiser, 1987; Loy, 1990]

(3) 시작품(Prototype)의 이용 : 새로운 소프트웨어를 개발할 때 사용자의 요구에 따른 완벽한 요구 명세서의 작성은 어렵다는 것이 쉽지 않다는 것은 미국 육군에서 소프트웨어 프로젝트를 조사한 결과에 의하면 47%가 사용불가, 29%는 연구비만 지급한 상태로 중단, 19%는 재작업, 단지 5%만이 그대로 사용가능하거나 수정 작업 후 사용할 수 있었음에서 알 수 있다.

소프트웨어 개발시 문제점중의 하나가 사용자의 요구정의가 정확하지 않음에서 비롯되므로 시작품 형태의 소프트웨어를 단기간에 제작하여 사용자에게 제시하여 요구사항을 확정하는 방법이다. 시작품 검토 시점에서 수정된 시작품을 만들어 가는 방법을 반복하는데 이를 위하여 시작품용의 프로그래밍 언어, 스크린의 LAYOUT, 보고서 설계를 위한 도구들이 필요하다.[Neghbor, 1984 ; Boar, 1984]

(4) 재사용 시스템 도구 CARS의 이용 : 중앙 대학교에서 연구 개발하여 현재 계속 보강중인 재사용 도구인 CARS는 유닉스/윈도우 환경하에서 C 와 C++를 기본 툴로 하고, 모델링과 객체지향 분석 및 설계기법을 표준화시켜서 작성된 클래스를 라이브러리에 저장하고 검색하도록 설계 하였다.[이경환, 1989]

그림 3의 CARS 구성에서 알 수 있듯이 사용자 인터페이스중에서 Finding Subsystem과 Building Subsystem은 별도로 지식기반 시스템으로 만들어 지원하고 있다. DB에는 재사용 가능한 부품들의 속성을 저장하며 저장된 모든 정보는 여러 가지 서브시스템에서 접근이 가능하고 DB를 보호하기 위해서 라이브러리 관리자에게만 DB내용을 수정할 권한을 준다. 라이브러리 관리자는 설계나 원시파일로 부터 재사용 정보를 추출하고 후보요소들의 품질을 보증하여 DB에 등록하고 관리한다.[CARS메

뉴얼, 1992]

CARS는 사용자 인터페이스, 라이브러리 관리, 라이브러리 감독 서브시스템등의 3개의 서브 시스템으로 구성되어 있다. CARS는 부품을 개발하여 저장하고 저장된 부품이 비슷한 기능의 것일 때는 이를 변경하여 사용할 수도 있는, 즉 저장된 부품에 새로운 기능의 추가, 삭제, 변경이 자유롭게 도와준다. 또한 저장된 부품을 검색 할 때도 특별한 질의어가 있는 것이 아니고 메뉴에 의한 사용자 인터페이스 이므로 질의어에 대한 이해와 구문의 암기등과 같은 부담이 없다. 사용자 인터페이스 서브 시스템은 사용자가 쉽고 빨리 원하는 부품을 찾아 재사용 하도록 편집과 질의처리등의 기능을 하며, 부품의 개발, 분류, 등록, 삭제, 갱신, 리스팅등의 라이브러리 관리 유지 기능은 라이브러리 관리 서브 시스템이 담당하고, 부품의 이용상황을 기록유지하는 로그 파일에 접근하여 부품별 사용빈도수를 산출하거나 라이브러리 안이 라이브러리를 관리할 때 참조할 수 있는

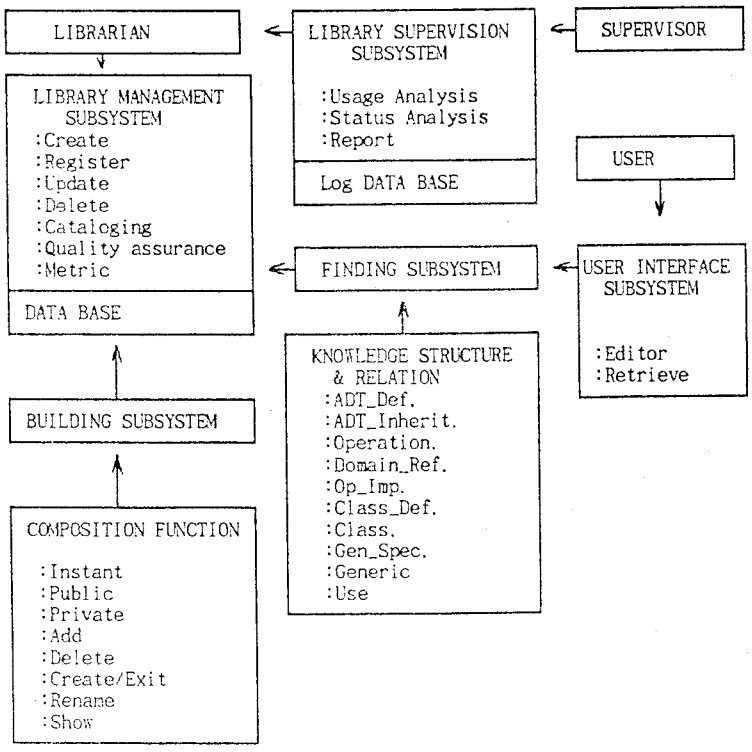


그림 3. 재사용 시스템 도구인 CARS의 구성

정보들을 얻기 위해서 라이브러리 상태분석등의 기능을 수행하는 라이브러리 감독 서브 시스템이 있다. 부품을 검색할 때는 해당 부품의 특별한 속성을 지정하거나 "I Need Stack Package"와 같은 자연 언어를 사용 할 수 있다.

(5) 소프트웨어 공장(Software Factory)의 이용 : 소프트웨어 공장의 효시는 일본의 '히타치'라고 하는 것이 일반적이며, 미국의 System Development Coporation 의 소프트웨어 공장, IBM의 Santa Teresa Laboratory로 발전하여왔다. 소프트웨어 공장은 소프트웨어를 제작할 제작팀이 관리체계를 정비하고 사용될 도구나 기법을 표준화 한 다음에 제작공정을 점진적으로 자동화 한다. 소프트웨어 공장은 일본에서 시그마 프로젝트를 착수한 것이 계기가 되어 ESPRIT 프로젝트의 SFINX, ESF의 EASF등이 추진되었다.[Matsumoto, 1986 ; Cusman, 1989]

Ⅲ. APRS의 설계 및 구현

1. APRS의 개요

APRS의 설계 목적을 크게 나누어 네가지로 구분 해보면, 첫째로, 소프트웨어 개발시에 가장 문제가 되는 사용자와 개발자간의 의사소통의 원활화로 시스템 요구 명세서 작성의 정확성을 획기적으로 높이는 것이며, 둘째, 기존의 재사용 방법과 기법에 의한 재사용 도구를 개선하는 것이고, 세번째로 재사용 도구의 적용분야에 응용업무 분야를 추가하고, 마지막으로 프로그래밍시에 범용언어를 사용할수 있도록 확장하는 것이다. 기존의 도구를 사용할 경우 생성될 프로그램의 기능과 특징들을 조기에 확인하기가 어렵고, 적용분야도 일반적인 응용업무 보다는 특정한 업무위주로 되어 있다. 지원가능한 단계도 소프트웨어 수명주기 전 단계를 지원하는 것이 아니라 설계단계나 코딩단계와 같이 일부분을 지원하는데 그치고 있다. 소프트웨어 재사용 대상을 원시코드에 국한시켜 원시코드의 재사용을 지원하는 시스템에 관하여서는 많은 기존의 연구가 있었다. 원시코드의 재사용을 위한 방법으로는 필요한 부품의 획득과 이해, 유사한 부품인 경우의 수정방법, 적합한 부품의 결합, 완성된 재사용 부품의 저장을 지원하는 시스템의 시작품이 시험 가동중이다. 그러나 원시코드의 재사용은 얼마간의 이득이 있기도 하지만 소프트웨어 수명주기 전체로 보았을 때는 그 비중이 20%에 불과하여 주기중 일부만 지원하므로 효과의 한계가 있고, 또한 원시코드 자체가 너무 특정영역에 대해서만 구체화가 된 상태이고 적용영역이 제한적이므로 코드 재사용의 이익 또한 제한적이다. 또한 이 도구들은 모두 텍스트에 기반을 두고 있으므로 편의성이 부족하고 설계정보의 분류나 검색, 기존 방법론과의 연계방안이 미비하다.

원시코드 재사용의 단점을 보완하여 모듈 스펙의 재사용을 추구하기위한 도구로 그래픽 에디터와 전 후진 검색기를 채용한 설계정보 재사용 시스템을 제안하고 있으나 사용자 입장에서는 원시코드의 재사용 보다 향상된 점을 느

낄수 없다. 그 외에도 효과적인 재사용을 지원하기위한 소프트웨어 부품의 분류에 관한 연구, 부품의 합성과 릴레이션 쉽에 관한 연구, 클래스 라이브러리 재구성에 관한 연구등 객체지향의 코드 재사용 시스템과 관련한 분야의 연구가 활발히 추진 중이다.[김치수, 1990 : 김행곤, 1991 : 변상용, 1990 : 진영택, 1991] 이러한 일련의 연구는 유닉스나 DOS 환경하에서 퍼스날 컴퓨터 위주의 시스템 도구로 범용언어에 의한 응용업무 위주의 시스템에 적용하기는 어렵다.

제안하는 응용 프로그램 재사용 시스템은 운용면에서 두개의 논리적인 구조를 갖는데 하나는 컴퓨터에 대한 기본지식을 갖춘 전문인을 위한 서브시스템 구조이고 또 하나는 전문인을 제외한 모든 일반 사용자가 이용할 수 있는 정보의 조회나 확인을 위한 일반 사용자 서브시스템 구조이다. 전문인이란 주로 전자계산소에 근무하는 전산인과 각 부처나 단과대학, 대학원등에서 직접 자료를 입력하고 수정하며, 그 자료의 생성과 삭제에 대한 책임이 있는 사람을 말하며 일반 사용자란 그외의 모든 대학구성원을 말한다. 응용 프로그램 재사용 시스템을 설계 및 구현하고자 할 때 가장 우려되는 부분은 자료의 보호와 보완에 관한 문제를 여하히 해결하는가 이다. 자료의 보호를 위해서는 자료를 일반 사용자용과 전문인용으로 분리 저장하여 일반 사용자는 전문인 영역에 접근하지 못하도록 하고 일반 사용자 접근가능 영역에도 교수, 교직원, 학생등의 신분에 따라 필요로 하

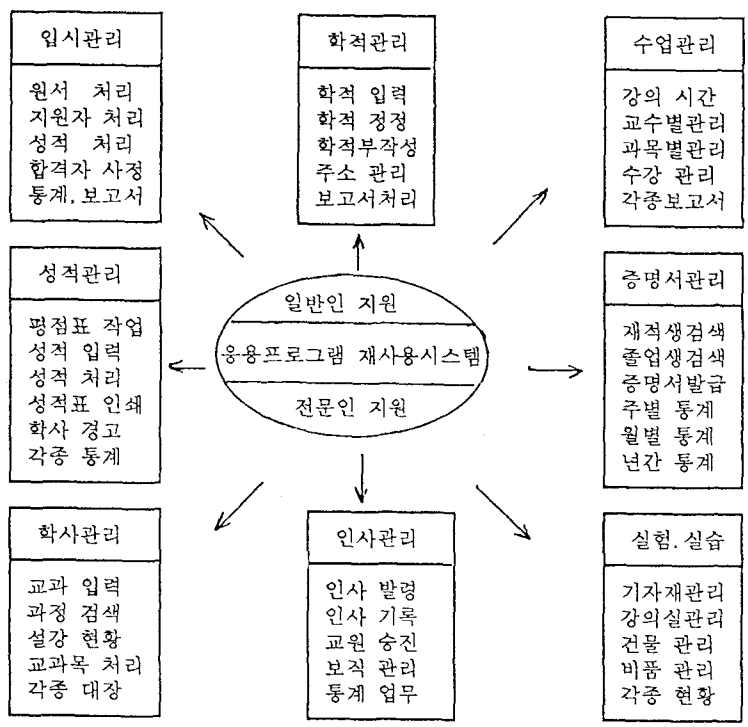


그림 4. 응용 프로그램 재사용 시스템 구조

는 정보에만 접근하게 한다. 전문인용 자료에 접근하는 경우에도 개인별의 비밀 인식표를 부여하고 사용전에 이를 확인하는 절차를 거치도록 한다. 이러한 자료보호를 위한 조치이외에 자료를 보완하는 방편으로 일반 사용자도 필요시 주별 백업과 월별 백업등을 병행하여 자료의 무결성을 보장한다.

APRS의 적용업무 분야는 대학 행정의 일부인 입시관리 업무와 신입생에 관련된 등록업무의 일부이다.

설계 및 구현된 시스템은 응용 프로그램 위주의 일반 사용자들이 쉽게 접근하고 재사용할 수 있는 도구이면서 소프트웨어의 개발을 담당하는 프로그래머를 도와주는 두가지 기능을 모두 수행하고 사용자의 요구사항을 분석하여 명세서를 작성할 때 발생하는 착오를 최소화 하기위해서 사용자가 소프트웨어 산출물을 바로 확인하는 사용자용의 확인 및 검색 시스템을 설계하고 구현된 시스템을 실행시켜 그 결과를 확인한다.

사용되는 하드웨어 시스템은 PRIME-4050 이며, 운영체제는 PRIMOS이다. 설계하고자 하는 재사용 시스템은 일반 사용자의 편의에 중점을 둔 부품확인 서브시스템으로 구성되는 바, 이들 서브 시스템은 FORMS와 SPOOLER등을 이용하고 사용할 언어는 COBOL이다. 재사용 시스템의 적용업무를 세부적으로 보면,

대학의 입시처리에 관한 업무로 하였고 재사용 가능한 모듈은 자료처리 구분에 따라 8개의 영역, 즉 원서 처리, 지원자 처리, 수능성적 처리, 합불 사정 처리, 합격자 처리, 등록금 처리, 통계 처리, 보고서 처리등이고 구현한 전체 모듈수는 170개이며, 각 각의 모듈은 최소길이가 54 라인이고 최대길이는 1062라인이다. 모듈을 분류한 기준은, 첫째, 모듈은 입출력, 명령어,논리적 처리절차 및 자료구조를 갖는다. 둘째, 모듈은 개별적으로 컴파일 할 수 있으며, 라이브러리 내에 저장 할 수 있다. 셋째, 모듈은 하나의 프로그램내에 내장 될 수 있다. 넷째, 모듈 세그멘트는 하나의 명칭 및 여러 개의 매개변수를 호출함으로서 사용 될 수 있다. 마지막으로 모듈은 다른 모듈을 사용 할 수 있다.

대학의 입시처리 시스템의 소프트웨어는 일

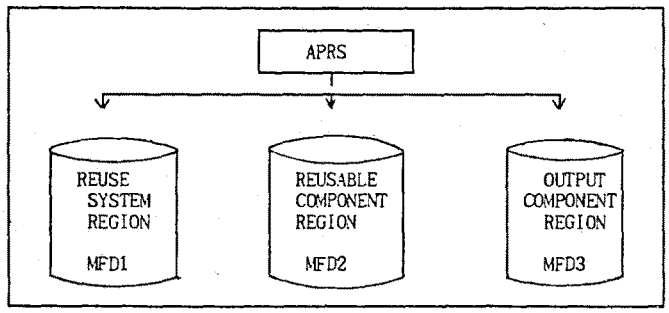


그림 5. APRS의 논리적 구성도

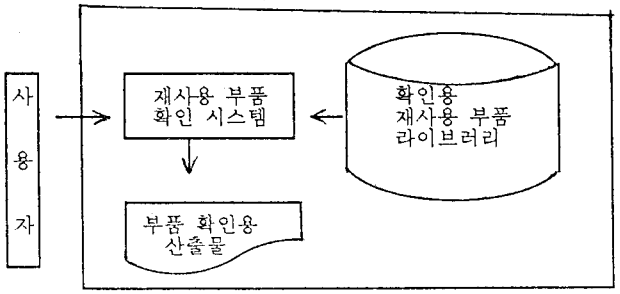


그림 6. 재사용 부품 확인 서브시스템 구조

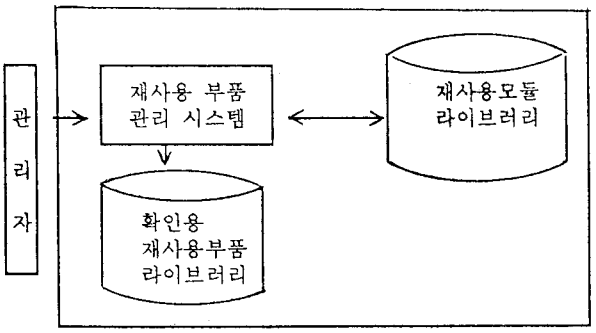


그림 7. 부품 관리 서브시스템의 구조

반적인 여타의 소프트웨어와 달리 고도의 신뢰성이 요구되고 소프트웨어의 규모가 비교적 크며 하드웨어의 고장이나 오 동작등에 대한 대비가 되어야 하고, 시스템의 수명이 길고, 시스템의 유지·보수작업은 전체적인 시스템의 교체 보다는 일부 기능의 변경만으로도 가능하다는 특징을 가진다.

2. APRS의 서브 시스템

응용 프로그램 재사용 시스템의 논리적인 구성도는 그림 5와 같이 크게 세가지로 나누어진다.첫째, 일반 사용자를 위한 라이브러리 구조로 주로 조회나 확인용의 자료나 정보로 구성된다. 둘째, 전문인을 위한 라이브러리 구조로 재사용 소프트웨어 부품을 생성하고 저장하며 이를 관리한다. 셋째, 응용 프로그램 재사용 시스템의 소프트웨어 부품을 저장하는 구조로 되어있다. MFD(Master File Directory)1에는 재사용 시스템에 관한 저장, 검색, 수정등의 모든 내용이 포함되는 영역이고 MFD 2의 재사용 부품 관리 영역에는 전문인을 위한 저장, 검색, 수정등의 모든 내용을 포함하며, MFD3에는 일반 사용자가 조회하고 확인하는데 필요

한 모든 자료와 정보를 포함하고 있다. 응용 프로그램 재사용 시스템은 사용자 측면에서 보면 두 종류이며 그 하나는 시스템이 보유한 각종 정보와 이 정보를 산출하는 해당 소프트웨어를 재사용 부품으로 생각할 때 이 부품을 사용하는 일반 사용자와 다른 하나는 이 부품을 생산하고 관리하는 개발 및 관리자의 두가지로 구분한다.

대학을 예로들면, 대학의 일반 사용자는 관리층과 교수, 행정·사무직원, 학생등이고 개발 및 관리자는 전자계산소에 보직된 시스템 분석가, 프로그래머와 같은 전산전문인과 행정사무를 담당하는 각 처부에서 응용업무를 전산화하는데 기능명세등을 작성 하는 등의 응용업무 개발 전담의 전문인등이다. 대학의 입시 처리 전산화 과정에서는 교무처와 같은 입시 담당 부처의 책임자와 담당직원이 일반 사용자이고 이들을 도와 실제 전산화를 해야하는 전자 계산소 개발 담당자들이 개발 및 관리자가 될 수도 있다.

설계하고자 하는 재사용 시스템은 두개의 서브시스템으로 구성되어 있다. 그 중 하나는 일반 사용자용으로 소프트웨어 개발에 관한 기본 지식이 없는 사용자가 필요로 하는 소프트웨어 부품을 검색하고 찾은 부품이 과연 사용자의 요구에 알맞은가를 곧바로 확인할 수 있는 부품검색 서브 시스템과 산출물 확인용 도구로 구성된 사용자 전용의 서브시스템이고 다른 하나는 재사용 소프트웨어 부품을 개발하고 이를 검증하여 고품질임이 판명된 부품을 라이브러리에 저장하고 이를 관리하는데 필요한 도구들로 구성된 소프트웨어 재사용 부품 관리 서브시스템이다.

그림 6은 사용자 전용의 재사용 부품 확인용 서브시스템의 구조도이다. 사용자는 부품 확인용 서브시스템의 메뉴에서 자신이 원하는 기능을 수행할 부품을 선택하면 재사용 부품 라이브러리에 저장된 확인용 산출물이 출력되고 적합한 경우에는 그 부품을 이용하며, 적합하지 않은 경우에는 그 출력물에 의하여 시스템 개발자와 의사소통의 수단으로 활용하여 모듈의 수정을 요구한다. 이와 같이 사용자가 직접 출력물을 초기에 확인할 수 있으므로 개발자와 사용자간의 의사소통이 원활하여 시스템 개발 기간을 단축 할 수도 있고, 출력물 자체를 이용하여 수정부분을 확인 할 수 있어서 기능명세가 한층 간편하고 정확하다.

재사용 소프트웨어 부품을 생산하고 이를 관리하는 부품 관리 서브시스템은 프로그래머가 자신이 소프트웨어를 개발할 때 이를 재사용 할수도 있고, 생성된 부품을 일반 사용자에게 지원을 하는 시스템으로써 구조도는 그림 7과 같다.

3. 재사용 시스템의 전체 자료의 흐름

대학 입시의 전산처리 과정을 자료처리의 흐름을 기준으로 단계별로 구분하면 그림 8과 같이 원서를 처리하여 원서 마스터 파일을 생성하는 단계와 원서 파일을 근거로 각종 고사점수와 내신점수 항목, 합격여부 항목등을 추가한 지원자 마스터 파일을 생성하는 단계, 지원

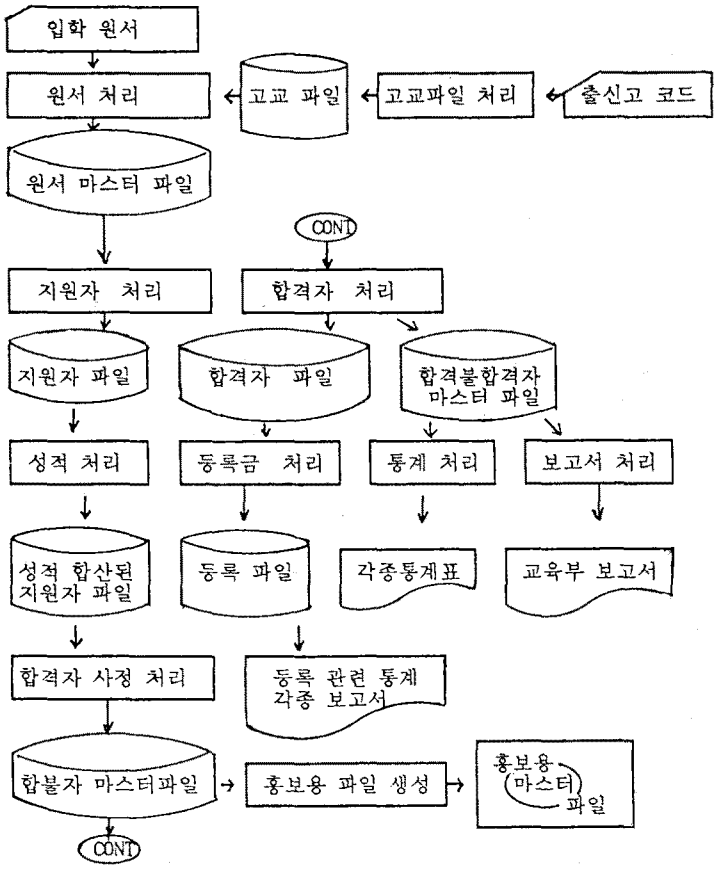


그림 8. 재사용 시스템의 전체 자료 흐름도

자 마스터 파일에 각종 고사성적을 갱신하고 과목별 득점을 합하여 총점을 계산하는 성적 처리 단계, 내신과 고사점수를 합산한 총득점에 의하여 합격자와 불합격자를 구분하는 합격자 사정 단계, 사정단계에서 합격자로 선발된 학생에 대해 합격자 마스터 파일을 생성하는 합격자 처리단계가 있다. 이후의 단계는 합격자 처리 단계에서 생성된 합격자 마스터 파일에 근거하여 등록금 마스터 파일을 생성하는 등록금 처리 단계와 합격, 불합격 파일을 이용하는 각종 통계 처리단계와 교육부 보고서를 중심으로 각종 보고서를 작성하는 보고서 처리 단계이다. 이들 단계의 각 업무는 다음과 같다.

첫째, 원서처리 단계에서는 원서 마스터 파일을 생성한다. 이 단계에서 맨 처음으로 할 작업은 국립교육 평가원에서 발행한 전국 고등학교 번호 일람에 의하여 고등학교 코드와 고등학교명을 포함한 고교코드 마스터 파일을 생성하는 일이다. 이 작업이 끝나면 확인용 고교코드 대조 리스트를 작성하여 원서 입력 이전에 고교코드 마스터 갱신작업을 마쳐야 한다. 원서 접수가 시작되고 전산소에 원서가 도착하면

각 대학의 형편에 따라서 원서 입력 방식이 결정되는데 일반적으로 메뉴에 의한 입력방식과 OMR에 의한 입력방법, 혹은 이 두가지를 혼용하는 방식이 있으며 제안하는 응용 프로그램 재사용 시스템에서는 두가지 방식을 혼용하는 시스템으로 구현 하였다.

두번째, 지원자 처리 단계에서는 이전 단계에서 생성한 원서 마스터 파일에 의하여 지원자 마스터를 생성하고 지원자의 각 항목의 이상유무를 확인하기 위한 대조명부를 작성하여 파일의 신뢰성을 확인하고 접수번호와 수험번호 대조명부를 작성한다.

세번째, 성적 처리 단계에서는 국립 중앙 교육 평가원에서 입수한 2차에 걸친 수학능력 시험성적 마스터와 각 대학별로 치루는 대학별 고사성적용 마스터를 생성하고 수능성적과 본고사성적, 내신 등급 별로 환산된 내신 성적을 합산하여 고사성적 마스터를 생성하고 성적의 신뢰성을 확보하기 위하여 각 시험별로 성적 대조 리스트를 작성 확인하고 각종 성적의 신뢰성이 확인되면 이를 변동자료로 하여 지원자 마스터를 갱신한다.

네번째 단계는 모집정원에 따라 각 대학별 각 학과별 혹은 우선전형, 특별전형별, 혹은 일반전형등의 전형구분에 따른 합격자와 불합격자의 사정을 하여 합격자와 예비 합격자(후보자), 불합격자를 구분한다.

다섯째, 합격자 처리 단계에서는 합격자와 후보자 마스터, 불합격자 마스터를 생성하는데 여기서도 이전 단계의 사정 방침에 따라 여러 종류의 합격자 마스터 생성 방식이 있는데 일반 전형이나 특별 전형, 우선 전형등 사정 작업시 생성된 파일의 종류와 파일 개수에 따라 합격자 마스터 파일을 생성하는 방식이 달라지며 제안된 재사용 시스템에는 이들 모든 방식을 지원하도록 구현하였다.

여섯째 단계는 이전 단계에서 생성한 합격자 마스터 파일에 의하여 등록금 마스터 파일과 장학생 마스터 파일, 고지서 파일을 생성하고 등록금 납입 통지서, 장학생 통지서등의 각종 통지서를 작성하여 발송할 수 있는 봉투까지 찍는다.

일곱째, 각종의 통계를 처리하는 단계이며 이 통계는 지원자와 합격자를 따로 작성하거나 지원자와 합격자를 대비하여 하나의 통계로 작성할 수도 있다. 합격자 처리 단계에서 생성된 합격자 마스터에 의하여 각종의 분포표나 통계표를 작성하는데 일반적으로 작성하는 분포표에는 내신 등급 분포, 수능성적과 본고사 성적 분포, 총 득점순의 분포등이 있고, 통계는 출신지별 혹은 선택 과목별의 통계와 출신 고교별, 1, 2 지망별, 연령과 성별 통계 등이 있다. 마지

기능 키	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
응용 영역	입시 관리	학적 관리	수업 관리	성적 관리	증명서 관리	학사 관리	인사 관리	실험실습 관리	메뉴 복귀	모든 작업끝

그림 9. 재사용 시스템의 주 메뉴의 기능키 구조

막의 보고서 처리 단계에서는 교육부에 보고할 각종양식에 따른 보고서와 수험생의 합격, 불합격여부를 전화국에 통보 하기위한 합불 확인용 마스터를 생성하고, 미등록생 발생에 따른 후보자의 합격자화 작업을 마치면, 합격자가 신입생이 되므로 합격자에게 학번을 부여하여 신입생 마스터를 생성한다.

Ⅳ. APRS의 실행

1. APRS의 시작과 종료

응용 프로그램 재사용 시스템은 F1에서 부터 F10까지의 기능키를 이용하여 설계된 형판(Template)을 사용하며 응용영역을 구분하는 주 메뉴의 기능키 구조는 그림 9와 같다.

응용 프로그램 재사용 시스템의 화면 구성은 크게 보아 제목부분, 본문, 제어, 메시지 부분의 4개 부분으로 되어있다.

제목 부분에는 업무영역과 처리구분, 처리일시등의 내용으로 세분 할 수 있다. 본문 부분에는 업무영역에 속하는 세부 기능 영역의 명칭과 이의 설명 부분으로 구성된다. 제어 부분에는 작업화면을 운영하는 각 기능키의 기능부여 내용을 나타내며 세부기능을 입력받아 작업을 수행한다. 메시지 부분은 작업화면에서 필요한 정보나 기능 수행상 필요한 정보들을 출력하며 사용자는 이 메시지에 의하여 작업수행에 도움을 받는다.

응용 프로그램 재사용 시스템은 대학을 예로 들면, 해당 대학내의 전체 교직원 및 학생이 재사용 시스템이 보유한 모든 정보나 자료를 이용 할 수 있도록 이를 구성하고 운영하며, 관리를 담당하는 시스템이다. 이 시스템은 대학 구

```
*** KCC CRT TERMINAL LYNX-3300B (KS 2BYTE HANGUL)     ***
***                                    REV. 6.50 1989.7.1 ***
login please
OK
LOGIN
User id ?
Password ?
Project id ?
```

그림 10. 재사용 터미널의 초기 화면

```
RUSE(user3) logged in Monday, 14 Feb 94 11:25:28.
Welcome to PRIMOS version 22.1.0
Copyright (c) 1989, Prime Computer Inc.
Serial #D02N-****DSTCPY* (Korea Computer Corporation)
Last login Saturday, 12 Feb 94 13:33:41
OK
```

그림 11. 재사용시스템의 RUSE 사용 허락 화면

성원 이외의 사용자가 시스템에 접근을 시도할 경우에는 "사용권한 없음"과 같은 화면을 출력하여 필요한 조치를 한후에 사용여부를 결정한다. 전자계산소의 허락을 득한 외부 사용자를 포함한 모든 사용권자가 사용할 터미널에 전원을 공급하면 그림 10과 같은 화면이 나타난다.

사용자가 'LOGIN'이라고 입력하면 시스템은 'User id?' 라고 응답하며 사용자는 'User id' ? 항에 부여 받은 사용자명 'RUSE'를 입력하면 'PASSWORD ?'가 나타나고 해당 패스워드 인 'RUSEPASS'를 입력하면 그림 11과 같은 재사용 허락 화면이 출력된다. 여기에서 유의할 사항은 사용자명을 15분 이내에 입력하지 않으면 시스템은 자동으로 타임아웃이 되며, 패스워드가 화면에 나타나지 않아도 시스템의 이상이 아니라는 점이다.

'RUSE'는 사용자 이름이고, 괄호안의 번호는 운영체제에 할당된 사용자 번호이다.

만약에 사용자 이름이나 패스워드를 잘못 입력한 경우에는 'Invalid user id or password; please try again.' 이 화면에 나타나며 이때는 사용자명과 패스워드를 다시 입력하고 프로젝트명을 잘못 입력하면 'Invalid Project id; please try again'이라는 메시지가 나오며 여기서는 프로젝트명을 재차 입력한다. 만약 사

```
A.P.R.S.                    A.P.R.S.   M A I N  M E N U          DATE : 94/02/17
R O O T                                                          TIME : 09:05 AM

      REGION ID              FUNCTION KEY          DESCRIPTION

      APRS110                   F1              입시  관리  SUB MENU
      APRS120                   F2              학적  관리  SUB MENU
      APRS130                   F3              수강  관리  SUB MENU
      APRS140                   F4              성적  관리  SUB MENU
      APRS150                   F5              졸업  관리  SUB MENU
      APRS160                   F6              증명  관리  SUB MENU
      APRS170                   F7              실험  관리  SUB MENU
      APRS180                   F8              인사  관리  SUB MENU
                                F9               RETURN APRS
                                F10             RETURN PRIMOS
                              FUNCTION KEY를 눌러 주시오.

MESSAGE :
Application Program Reuse System                SCREEN-ID : APRS100
```

그림 12. 응용 프로그램 재사용 시스템의 초기 화면

```
RUSE  (user 3) logged out Wednesday,  16, Feb 94 14:30:48.
Time used : 01h 14m connect,  120m 06s CPU,  148m 56s I/O.
```

그림 13. 작업 종료 화면

용자명과 패스워드, 프로젝트명을 수차례 입력해도 계속하여 에러가 발생하면 시스템 관리자에게 문의한다.

정상적인 등록절차가 완료되면 모니터에는 'OK'가 다시 나타나며 여기에 'APRS'라고 입력하면 그림 12와 같은 응용 프로그램 재사용 시스템의 초기화면이 나타난다. 응용 프로그램 재사용 시스템의 접근 가능영역에는 APRS의 초기 화면에서 알 수 있는 바와 같이 기능키의 선택 여부에 따라 8개의 업무영역으로 구분되어 있다.

각 영역에는 각 영역별의 고유 업무들이 하위 영역으로 세분되어 존재하는바 하위영역을 기능키를 중심으로 나타내면 다음과 같다.

첫째, F1 기능키는 입시 관리 업무로 하위 메뉴에는 원서 처리, 지원자 처리, 성적 처리, 합격자 사정, 각종 보고서 처리가 있다. 둘째, F2 기능키는 학적 관리 업무로 하위 메뉴에는 인사관리와 급여관리, 시설관리등을 포함한 5개의 업무영역으로 세분이 된다. 셋째, F3 기능키는 수업 관리 업무로 하위 메뉴에는 학생지도와 장학관리, 동아리 관리등을 포함한 5개의 세부 업무영역이 있고, 넷째, F4 기능키인 성적 관리 업무에는 강의실 관리와 수강신청, 교과목 관리등을 포함한 6개의 하위 업무영역이 있다.

그외에도 F5 기능키는 졸업관리, F6 기능키는 증명그림14. 입시관리 시스템의 주 메뉴서 관리 업무이며, F7 기능키는 실험관리, F8 기능키는 인사 관리 업무등의 하위영역의 기능키가 있다.

초기 화면의 업무영역 선택에 따라 사용자는 재사용 시스템을 이용할 수 있으며 시스템을 종료하고 싶을 때는 F10을 선택한다.

그러면 화면상에는 작업용 메뉴가 사라지고 운영체제상태로 환원되어 화면상의 좌상단에 'OK'가 나타나면 모든 작업을 끝내기 위하여 'LogOff' 혹은 'LO'를 입력하면 그림 13과 같은 화면이 나타나면서 재사용을 종료한다.

2. APRS 재사용 부품의 확인 및 검색작업

입시 관리에 관한 재사용 시스템을 이용하려면 먼저 그림 12의 응용 프로그램 재사용 시스템의 초기 화면에서 기능키 F1인 입시관리를 선택하면 입시관리 재사용 시스템 주 메뉴가 화면에 나타나며 8개의 처리업무 영역별로 하위메뉴와 연결되어 있다. 주 메뉴의 본문 구성항목은 가로 방향의 행에는 재사용 가능한 하위 영역명과 그 하위 영역에 접근하는데 필요한 기능키의 할당, 해당 하위영역에 대한 설명순으로 배열되어 있고, 세로 방향의 열에는 확인 가능한 하위 영역의수 만큼의 하위 영역명들과 실제 기능키 할당내용, 해당 하위 영역의 설명란이 있다.

그림 14는 입시관리 시스템의 주 메뉴이며, 구성항목은 하위의 업무영역을 한눈에 알아 볼 수 있도록 각 업무를 처리하고자 하는 단계를 기준으로 8개 분야로 세분하고 사용자는 선택하고자 하는 하위영역만 지정하면 선택된 업무영역으로 연결시켜 준다. 이 단계에 이르면 각 사용자는 자신이 필요로 하는 항목의 그림 15.

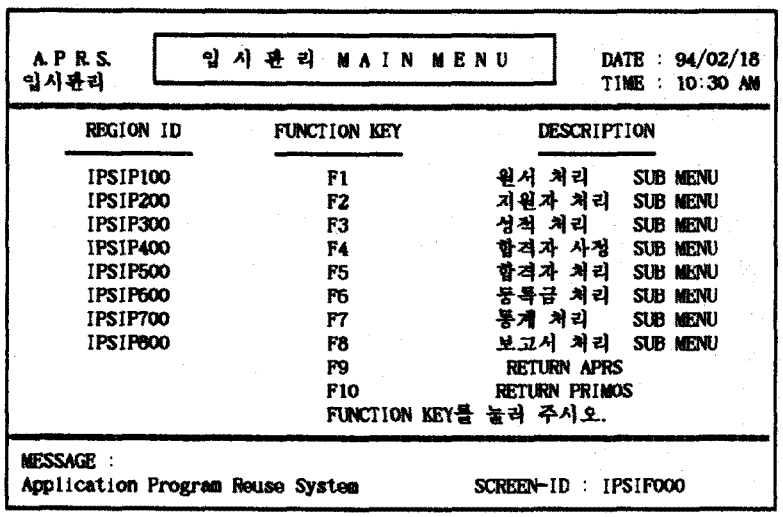


그림 14. 입시관리 시스템의 주 메뉴

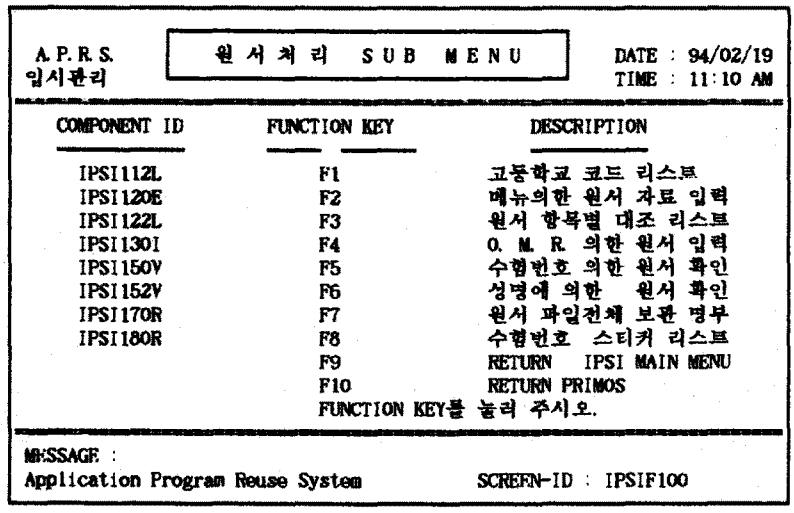


그림 15. 원서 처리 서브시스템의 메뉴

원서 처리 서브시스템의 메뉴정보를 얻기 위하여 8개의 업무 영역 중 하나를 선택 하여야 한다. 여기서 선택할 수 있는 하위영역은 원서처리와 지원자 처리, 성적처리와 합격자 사정, 합격자 처리와 등록금 처리, 각종 통계와 보고서 작성 영역이다. 하위 영역 8개 단계별로 사용자가 부품을 확인하는 절차는 매우 간단하며, 그 방법은 확인 하고자 하는 처리단계의 기능키를 선택하여 누르기만 하면 응용 프로그램 재사용 시스템이 다음 동작을 안내한다.

2.1. 원서 처리 단계의 실행

원서처리 단계를 실행하기 위하여 입시관리 주 메뉴에서 원서처리 서브 메뉴인 F1을 선택하면 그림 15와 같은 원서처리 서브메뉴가 화면에 나타난다.

이 단계에서 확인 할 수 있는 부품 정보에는 전국 고등학교 코드 리스트, 원서 입력을 위한 메뉴의 형식, 원서의 각 항목별 대조 리스트, 수험번호나 성명에 의한 항목 확인, 스티커 리스트등이 있다. 원서처리 서브시스템의 메뉴에서 전국 고등학교 코드 리스트를 확인하려면 기능키 F1을 선택한다.

원서를 입력하는 방식에는 원서자체를 OMR.로 처리하거나 혹은 터미널 상에서 화면을 통하여 직접 입력 할 수가 있으며 후자의 방식으로 직접 입력하는 경우에 해당 부품을 확인 하려면 원서처리 서브메뉴에서 기능키 F2를 선택한다.

```
입 시 관 리                    원 서 자 료 입 력          D A T E : 94/02/26
처리코드 :                                              처리구분: A=신규 D=삭제
접수번호 :                                                        R=변동 X=휴식
원 서 처 리                                             수험번호 :

*성    명:                *주민번호:          -
*제1 지망:              *주야구분1:              (코드 : 1=>주간, 2=>야간)
*제2 지망:              *주야구분2:              (코드 : 1=>주간, 2=>야간)
*졸업년도:        *고교코드:              *고교명:                    고등학교
*석차 백분율:    *결석일수:    일        *특별활동:     ( 1->5등급으로 입력)
*산업구분:  (코드 : 1=>산업동일, 2=>직무자격, 3=>자격 혹은 직무, 4=>해당)
*수능시험 차수:    *수능시험 수험번호:                    *수능시험 총득점:
*검정고시 내신등급:
*정원외 구분:   (코드: 0=>해당무 1=>전문 2=>학사 3=>예박 4=>산업 5=>외교)

***  F1 = 입력  F2 = 재실행  F3 = 수정  F9 = 주 메뉴 복귀  F10 = 작업 끝
 MESSAGE :
 Application Program Reuse System  SCREEN-ID          : IPSIF120
```

그림 16. 원서 자료 입력 화면

```
입시 관리                      원서 성명 조회          DATE : 94/03/02
원서 처리                                              TIME : 10:00 AM
성명 :                                                 총 인원수 :

| 일련번호 | 학 과 | 주민 등록 번호 | 특별여부 | 접수번호 | 비 고 |
|---|---|---|---|---|---|
|  1 |  |  |  |  |  |
|  2 |  |  |  |  |  |
|  3 |  |  |  |  |  |
|  4 |  |  |  |  |  |
|  5 |  |  |  |  |  |
|  6 |  |  |  |  |  |
|  7 |  |  |  |  |  |
|  8 |  |  |  |  |  |
|  9 |  |  |  |  |  |
| 10 |  |  |  |  |  |

F1=ENTRY   F2=FORWARD   F3=BACKWARD   F4=RESTART   F10=STOP

 MESSAGE :
Application Program Reuse System SCREEN-ID          : IPSIF152
```

그림 17. 원서 성명 조회 부품

그림 16은 화면상에서 직접 원서를 입력하는 원서 입력용의 부품으로 화면구성을 보면 제목부분 업무영역을 표시하는 항목에 주 처리 업무영역인 '입시처리' 와 하위 업무영역으로 '원서 자료 입력'을 명시하고 '수험번호', '처리 연월일' 항목이 있고, 신규 입력인가, 혹은 변동처리와 삭제인가, 휴식인가를 구분하여 입력하는 처리코드 항목과 지원자의 접수번호를 입력하는 항목으로 구성되어 있다.

본문 내용을 살펴보면 본문에서의 출력항목은 출신 고등학교명과 수학능력시험의 총점 항목이고 입력해야할 항목은 본문 내용 중 위의 두개항목을 제외한 성명에서 정원외 구분까지의 16개 항목이다. 제어부분에는 각기능키의 고유기능을 나타내는데 기능키 F1은 지원자의 원서 내용을 최초로 입력하고자 할 때 선택하고, 기능키 F2는 같은 일을 반복 재 실행할 때, F3 기능키는 기존에 입력 저장된 원서 내용 중 수정할 부분이 있을 때 선택한다. 기능키 F9는 입시관리의 주 메뉴로 복귀하고자 할 때, 기능키 F10은 원서처리 작업을 종료하고자 할 때 선택한다.

메뉴에 의한 입력 및 처리에서는 생성된 원서 마스터에 의하여 원서의 각 항목을 터미널 상에서 입력하고 수정하며 OMR에 의한 입력시에는 읽어들인 일반파일을 랜덤파일로 변환하는 단계를 거치는데 이때에도 수정항목을 OMR 용지상에서 수정하고 재차 읽힐 것인지 아니면 에러 자료를 포함하여 읽힌 다음 원서 마스터를 생성한 후에 수정할 것인 지의 두가지 방식을 선택 해야한다.

원서에서 동명이인(同名異人)을 확인하는 부품은 기능키 F6을 선택 해야하며 동명이인의 부품내용은 그림 17과 같다.

성명에 의한 원서 확인 부품의 구성 항목은 기본키가 되는 성명과 성명에 해당하는 인원이 몇명인가, 총 인원수를 출력하는 제목 부분과 각개인의 인적사항을 출력하는 본문부분, 처리를 제어하는 기능키 설명부분과 사용자에게 메시지를 전하는 제어부분으로 되어있다. 여러번의 대조와 수정작업을 반복하여 입력된 원서 자료자체에는 이상이 없다는 것이 확인되면 원서 마스터 파일을 생성한다. 원서 마스터 파일이 생성되면 수험번호를 부여하는데 수험번호 부여시 사용할 분류 키를 주민 등록 번호로 하는가 성명으로 하는가 이들의 조합으로 할 것인가에 따라 분류방식이 달라지는바 설계한 재사용 시스템에서는 두가지 방식 모두를 포함하고 있다.

원서의 각 항목내용을 수험번호에 의하여 확인하려면 기능키 F5를 선택한다. 수험번호에 의한 원서 각 항목의 확인용 부품의 화면 내용은 주어진 수험번호에 해당하는 지원자의 원서 각 항목으로 되어있다. 원서 확인 및 수정용 부품의 구성 항목은 기본키가 되는 수험번호와 보조키인 접수번호, 확인인가, 수정인가의 처리구분을 결정하는 처리구분 코드를 나타내는 제목부분과 각개인의 인적사항과 지원사항을 망라한 본문부분, 각각의 기능키의 기능을 설명한 기능키 설명부분, 각종의 제어를 담당하는 메시지 부분을 포함한다.

2.2. 지원자 처리 단계의 실행

대학의 실정에 따른 본고사 실시여부가 이 단계의 작업량을 결정하는 변수로 작용하나 일반적으로 해야하는 고사 준비 작업에는 면접고사 점검표와 실기고사 혹은 예.체능 특기자에 대한 특기고사 점검표, 신체검사 점검표등을 작성한다. 확인용 명부로는 고사장, 고사실 확인용, 검정자나 직업 훈련생, 정원외 입학 대상자들의 명부와 대장 작업으로는 수험순 전형대장, 내신 성적순 전형대장, 지원자 전체의 보관용 대장등이 있다. 각종 점검표에 의한 결시자의 갱신이 끝나면 지원자 마스터는 성적을 제

```
A.P.R.S.                  지 원 자  처 리  S U B  M E N U          DATE : 94/03/03
입시관리                                                           TIME : 12:10 AM

   COMPONENT ID          FUNCTION KEY          DESCRIPTION

     IPSI210R               F1          고 사 실 용      좌 석 표
     IPSI211R               F2          면 접  고 사     점 검 표
     IPSI214R               F3          신 체  검 사     점 검 표
     IPSI220R               F4          수험번호 순  지원자 대장
     IPSI230R               F5          면 접   결 시 자   명부
     IPSI250V               F6          수험번호의한 지원자 확인
     IPSI252V               F7          성명에 의한 지원자 확인
     IPSI280R               F8          지원자 전체 보관용 명부
                            F9          RETURN   IPSI MAIN MENU
                            F10         RETURN PRIMOS
                            FUNCTION KEY를 눌러 주시오.

MESSAGE :
Application Program Reuse System                SCREEN-ID : IPSIF200
```

그림 18. 지원자 처리 서브시스템의 메뉴

```
입 시 관 리                                                  D A T E : 94/03/07
지원자 처리                  지원자  확인  처리              처리구분: A=신규 I=조회
처리코드:                                                              R=변동 D=삭제
수험번호 :                                                   접수번호 :

*성    명:                *주민번호:
*합격학과 :                *지망구분:                (코드 : 1=>1지망, 2=>2지망)
*합격여부:          (코드 : U, H, J, *=>합격, 공란=>불합격)   *합격,주야구분:
*특채구분:          (코드 : 1=>산업동일, 2=>직무자격, 3=>자격 혹은 직무, 4=>해당없음)
*제1 지망:          *주야구분1:               (코드 : 1=>주간, 2=>야간)
*제2 지망:          *주야구분2:               (코드 : 1=>주간, 2=>야간)
*졸업년도:          *고교코드:                *고교명:                    고등학교
*석차 백분율:       *결석일수:     일         *자격코드:     ( 숫자 5자리로 입력)
*정원외 구분:  (코드: 0=>해당무 1=>전문대 2=>학사 3=>군위탁 4=>산업체 5=>외교관)
*수능시험 차수:          *수능시험 수험번호:                    *수능시험 총득점:
*교과시험 결시:  *면접고사 결시:  *신체검사 결시:               *결시 구분:
*교과내신등급:   *출석내신등급:   *특별활동 내신등급:          *검정고시 내신등급:
*교과환산점수:   *출석환산점수:   *특별활동 환산점수:          *내신환산점수 소계:
*본고사 득점:    *예.체능고사점수:  *특기고사 점수:            *고사점수 소계:
*시험성적 합산 총점:               *학과별 순위:

***  F1 = 신규입력  F2 = 재실행  F3 = 수정요망  F9 = 주 메뉴 복귀  F10 = 작업 끝
 MESSAGE :
 Application Program Reuse System              SCREEN-ID : IPSIF250
```

그림 19. 수험번호의 한 지원자 확인 부품

외한 모든 갱신이 완료된 파일이 된다. 면접 결시자, 신검, 실기, 특기 고사 별로 결시자 명단을 작성한다.

입시관리 주 메뉴에서 지원자 처리 단계로 접근하려면 기능키 F2를 선택한다. 지원자 처리 단계의 서브메뉴는 그림 18과 같으며, 확인 가능한 부품의 종류는 고사실용 좌석표를 비롯하여 면접이나 신체검사의 점검표, 수험번호순 지원자 대장과 면접,신검의 결시자 명부, 보관용 명부등이 있고, 수험번호나 성명에 의한 지원자 확인용의 부품들이 있다.

수험번호에 의한 지원자 확인용 재사용 부품을 검색, 확인 하려면 지원자 처리 서브 메뉴에서 기능키 F6을 선택한다. 지원자 확인용 재사용 부품의 구성항목은 그림 19에서 보는바와 같이 수험번호를 기본키로 하고 접수번호를 보조키로 하여 성명, 주민번호, 졸업년도와 출신고교, 제 1 지망과 2 지망등의 원서 내용 항목과, 합격여부, 지망구분, 특채 구분등의 각종 구분 항목이 있다.

이 부품은 지원자나 합격자의 마스터 파일내용인 각 항목을 확인 및 조회를 할 수 있는 부품이며 필요시에는 수정을 할 수도 있다. 교과 내신성적 환산점수, 출석 및 특별활동의 내신성적 환산점수등의 환산점수와 각종 고사점수, 이들을 모두 합한 총득점 항목, 그리고 각종 시험에서 본고사나 면접고사, 실기고사나 특기고사, 신체검사등의 결시여부를 표시하는 결시표시 항목등으로 구성되어 있다.

본고사나 면접, 예·체능계의 실기 및 특기고사, 신체검사등을 실시하려면 각 고사장과 고사실 별로 책상에 수험번호가 포함된 좌석표를 부착하여야 하며, 좌석표에 대한 부품을 확인하려면 지원자 처리 서브 메뉴에서 기능키 F1을 선택한다. 좌석표의 실제 양식은 용지 각 1매당 좌우로 각각 2명씩 배열하고 상하로 3열로 배치하여 양식 1매에 6명을 할당 할 수 있도록 하였으며, 부품내용에는 고사장과 고사실의 번호, 수험번호, 성명,지망학과등이 포함된다. 면접고사 점검표용의 부품은 각 고사장, 고사실 별로 40명을 기본단위로 작성하며, 불 응시자의 표시방법과 고사 담당교수의 확인 날인항을 포함한다.

면접고사 점검표의 부품내용을 확인하려면 지원자 처리 서브메뉴에서 기능키 F2를 선택한다. 수험번호순으로 작성한 지원자 명부는 일부 대학에서는 지원자 수험번호순 전형대장으로 명명하고 있으며 지원자 처리 서브시스템에서 이 부품의 내용을 확인 하려면 기능키 F4를 선택한다. 면접고사 점검표는 하나의 양식을 조금씩 수정하여 신체검사 점검표와 실기고사 점검표, 특기자 점검을 위한 특기 점검표 등으로 변형하여 재사용이 가능하다. 수험번호순의 지원자 대장은 대학의 실정에 따라 지원자 전체에 관한 보관용 명부로 혼용이 가능하나 필요시에는 지원자 마스터 파일 내용을 그대로 인쇄용지에 옮겨 저장된 마스터 파일의 대용으로 쓸 수도 있다.

지원자 전체 보관용 명부의 재사용 부품을 확인 하려면 지원자 서브 메뉴에서 기능키 F8을 선택 한다. 지원자 전체 보관용 명부는 학과별로 구분하여 양식 한장에 20명을 기준으로

인쇄하며, 작성 연월일과 페이지를 매 장마다 표시한다. 부품에 포함된 항목은 지원자 마스터 파일의 내용중 수험번호를 기본키로 하고 접수번호, 성명과 성별, 생년월일과 고교 졸업년도와 출신학교, 면접과 신검등의 합격여부, 전형구분과 합격과 불합격 여부등을 포함하여 작성한다.

2.3. 성적 처리 단계의 실행

성적 처리 단계에서 재사용 가능한 부품의 내용은 각종 시험성적의 정확성을 확인 하기위한 성적 대조 리스트와 확인된 성적의 입력과

```
A.P.R.S.                    성 적 처 리   S U B   M E N U          DATE : 94/03/08
입시관리                                                          TIME : 09:10 AM

   COMPONENT ID          FUNCTION KEY          DESCRIPTION

      IPSI312L              F1                 본 고사 성적 대조 리스트
      IPSI330L              F2                 수학능력 성적 대조 리스트
      IPSI334R              F3                 통합 결시자 명부
      IPSI340U              F4                 본 고사 점수 입력
      IPSI342U              F5                 실기고사 점수 입력
      IPSI344U              F6                 수학능력 점수 입력
      IPSI380V              F7                 수험번호 의한 개별 점수 확인
      IPSI390R              F8                 총점의한 석차순 전형대장
                            F9                 RETURN   IPSI MAIN MENU
                            F10                RETURN PRIMOS
                            FUNCTION KEY를 눌러 주시오.

MESSAGE :
Application Program Reuse System                SCREEN-ID : IPSIF300
```

그림 20. 성적 처리 서브시스템의 메뉴

```
입시 관리                                                          DATE : 94/03/10
성적 처리                    수학 능력 시험 총점 입력              처리구분 A =입력
처리 코드 :                                                        I =조회 R =변동

| 순위 | 수험번호 | 성 명 | 총 점 | 순위 | 수험번호 | 성 명 | 총 점 |
|------|----------|--------|-------|------|----------|--------|-------|
|  01  |          |        |       |  11  |          |        |       |
|  02  |          |        |       |  12  |          |        |       |
|  03  |          |        |       |  13  |          |        |       |
|  04  |          |        |       |  14  |          |        |       |
|  05  |          |        |       |  15  |          |        |       |
|  06  |          |        |       |  16  |          |        |       |
|  07  |          |        |       |  17  |          |        |       |
|  08  |          |        |       |  18  |          |        |       |
|  09  |          |        |       |  19  |          |        |       |
|  10  |          |        |       |  20  |          |        |       |

F1 = ENTRY  F2= RESTART  F3= UPDATE  F8= PREMENU  F9= MAINMENU  F10= STOP
MESSAGE :
Application Program Reuse System.                SCREEN-ID : IPSIF310
```

그림 21. 수학능력시험 총점 입력용 부품

각종 시험의 결시자에 대한 처리와 결시자 명부, 성적을 합하여 최종적인 각 개인별 총득점을 환산하고, 그 총점에 의한 석차순의 지원자 명부를 작성하는데 필요한 부품들이다.

성적 처리 단계를 실행하려면 입시관리 주 메뉴에서 기능키 F3을 선택한다.

그림 20은 성적 처리 서브 메뉴 이다.

본고사 점수 입력용의 부품을 확인하려면 성적 처리 서브메뉴에서 기능키 F4를 선택한다. 본고사 점수 입력용의 부품은 본고사 과목의 종류에 따라 각 대학별로 실정에 맞게 조정할 필요가 있으며, 주관식과 객관식을 혼용한 경우에는 추가의 부품이 필요하다.

수학능력 시험성적을 입력하는 재사용 부품을 확인 하려면 성적 처리 서브 메뉴에서 기능키 F6을 선택한다.

그림 21은 수학 능력시험 점수의 총점을 입력하는 화면이다.

국립 중앙 교육 평가원에서 고등학교의 수학능력 시험성적을 획득하여 이를 각 개인의 총득점에 합산해야 하는데 수학능력 시험성적을 컴퓨터에 입력하는 과정에서 제도적으로 사람의 수작업이 개입되어 있어서 입력시 발생 할 수도 있는 오차를 최소한으로 줄일 방도를 찾아야한다. 수학능력 시험의 총점은 100단위에 소수점 한자리로 구성되어 있고, 입력용 한화면에 20명 단위로 입력이 가능하다. 수학능력 성적을 대조하는 리스트의 부품을 확인하려면 성적 처리 서브메뉴에서 기능키 F2를 선택한다. 수학능력 성적 대조리스트는 1매당 각 학과별 40명단위로 출력하고 출력항목은 테이프 점수와 컴퓨터에 입력된 점수를 비교하여 수능 성적의 신뢰성을 확인한다.

2.4. 합격자 사정 단계의 실행

합격자 사정단계의 부품을 확인하려면 입시

```
A.P.R.S.                  합 격 자  사 정  S U B  M E N U          DATE : 94/03/12
입시관리                                                          TIME : 14:10 AM

   COMPONENT ID          FUNCTION KEY          DESCRIPTION

     IPSI410C              F1                  야간 특별  전형
     IPSI412C              F2                  주간 특별  전형
     IPSI420C              F3                  야간 일반  전형
     IPSI422C              F4                  주간 일반  전형
     IPSI434C              F5                  제 2 지망  전형
     IPSI440C              F6                  후보 10 %  전형
     IPSI460C              F7                  우선(추천) 전형
     IPSI490C              F8                  1 지망 80% 전형
                           F9                RETURN   IPSI MAIN MENU
                           F10               RETURN PRIMOS
                           FUNCTION KEY를 눌러 주시오.

MESSAGE
Application Program Reuse System                SCREEN-ID : IPSIF400
```

그림 22. 합격자 사정 서브시스템의 메뉴

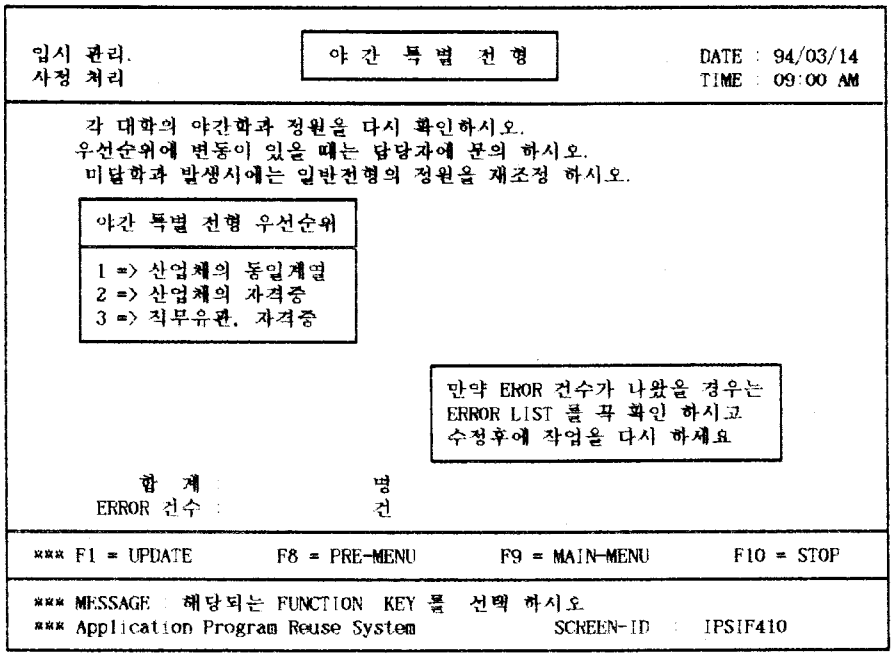


그림 23. 야간 특별 전형 부품

관리 주 메뉴에서 기능키 F4를 선택한다. 그림 22의 합격자 사정 서브 시스템에서 알 수 있듯이 이 단계에서 확인가능한 부품은 주·야간 특별전형과 주·야간 일반전형, 대학 특성에 따른 우선전형이나 80% 전형등의 부품이 있다. 합격 사정의 방침은 각 대학별로 상이하므로 일괄적으로 결정하기는 어려우나 일반적으로는 일반 전형과 특별 전형, 성적 우수자 선발을 목적으로 하는 우선 전형 혹은 1 지망에서 80 %만 선발하고 2 지망으로 나머지 20 %를 선발 하는 등이 있는바 제안하는 재사용 시스템에서는 두 가지 방안을 모두 수용하도록 구현하였다.

야간 특별 전형의 부품을 확인 하려면 합격자 사정 서브메뉴에서 기능키 F1을 선택 해야 한다. 그림 23은 야간 특별 전형의 부품으로 특별전형에 관한 우선순위와 미달시의 보충방법은 각 대학별로 상이하므로 실정에 알맞게 재조정 해야한다.

이 단계에서는 전형 별로 합격자와 불합격자를 사정 선발해야 하는데 각 대학별로 실정에 따라 약간씩의 차이가 있으므로 각각의 재사용 부품을 확인하고 각 대학에 알맞은 시스템으로 수정보완 해야한다. 제시한 야간 특별전형의 우선순위는 일부 대학의 경우이며 필요시 재조정이 가능하다. 주간 일반 전형에 관한 부품을 확인 하려면 합격자 사정 서브메뉴에서 기능키 F4를 선택한다. 야간 특별 전형과 달리 주간 일반 전형은 특별한 조건이 없이 지원자를 학과별, 총득점순으로 내림차순 정렬을 한 다음 모집정원 만큼씩 합격자 구분코드를 갱신하면 된다. 단지 유의할 사항은 입시전형 요강에 따라, 결시자나 합격선 이하의 득점자를 제외하고 정렬을 해야한다. 주간 일반전형에 관한 부품은 미달학과 발생시에는 제2지망자에 의하여 결원을 보충 하던지 다른 방안을 모색하던 지는

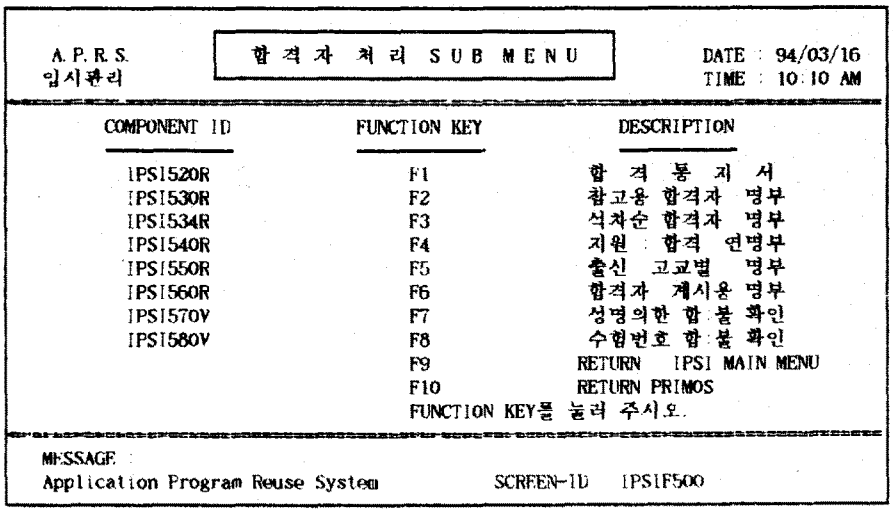


그림 24. 합격자 처리 서브시스템의 메뉴

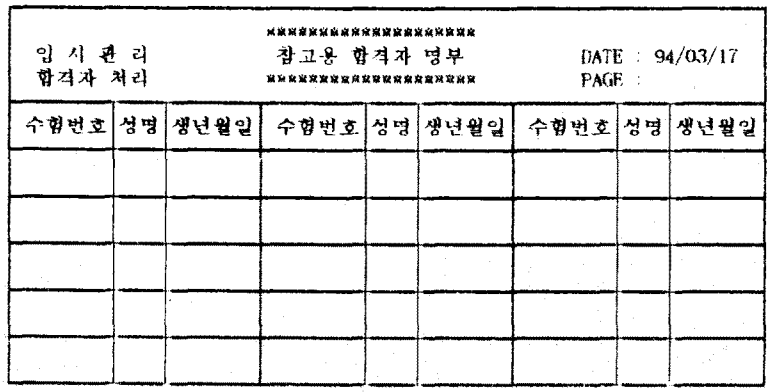


그림 25. 참고용 합격자 명단 부품

대학별로 부품을 재조정 해야한다.

2.5. 합격자 처리단계의 실행

합격자 처리단계의 부품을 확인 하려면 먼저 입시관리 주 메뉴에서 기능키 F5를 선택 해야 한다.

합격자 처리 단계의 각종 출력물에 대한 양식은 각 대학의 실정에 따라 알맞게 조정하는데 이때 준거가 되는 것이 바로 여기에서 확인하는 재사용 부품이 된다. 합격자 처리 서브 시스템의 메뉴는 그림 24와 같이 합격 통지서와 각종의 합격자 명부, 합격자를 계시 하기위한 방을 작성하고, 화면을 통해서 합격자와 불합격자를 확인한다.

참고용의 합격자 명부에 대한 부품을 확인하려면 합격자 처리 서브메뉴에서 기능키 F2를 선택한다.참고용 합격자 명부는 그림 25에서 알 수 있듯이 합격자의 수험번호와 성명, 생년월일등의 최소한의 항목만을 발췌하여 한면에 여러 명을 인쇄할 수 있도록 한다.

합격 통지서를 독립된 하나의 양식에 2명씩 출력하고자 한다면 합격자 처리 서브메뉴에서 기능키 F1을 선택한다. 합격 통지서는 합격통지서 자체를 하나의 독립된 양식으로 작성할 수도 있고, 합격통지서와 장학 통지서, 납입금 고지서, 납입금 영수증등을 하나의 양식으로 묶어서 작성 할 수도 있는데 이는 각 대학의 실정에 따라 조정 가능한 부분이다. 참고용의 합격자 명부도 양식 1매당 100명 단위에서 하나의 학과 단위의 인원으로 할 수도 있고, 포함할 항목도 학과, 수험번호, 성명, 생년월일등의 4개항으로 하던지, 생년월일 항은 제외 할 수도 있다.

출신고교 별 명부는 각 고등학교의 진학담당 선생님들에게 송부 하던지 아니면 해당 고등학교의 선생님께서 직접 가져가시기도 하는 각 고등학교에서는 꼭 필요로 하는 부품이다. 출신 고교별 명부의 부품을 확인하려면 합격자 처리 서브메뉴에서 기능키 F5를 선택한다. 출신 고교별 명부의 구성항목을 보면, 출신 고등학교명, 합격한 학과, 합격자용인가 후보자용인가의 구분, 각 개인의 인적사항으로 수험번호와 성명, 생년월일, 그리고 고등학교 졸업 연도등이 포함된다.

합격자 계시용 명부는 합격자 발표 당일에 계시하여 1차 등록 마감일 까지는 계속하여 게시판에 부착되어 있어야 하고, 동시에 여러 사람이 볼 수 있어야 한다. 그러나 일반적인 범용의 라인 프린터로 합격자를 인쇄할 경우 인행기 자체의 약점으로 인하여 확대가 불가능 하므로 합격자 마스터 파일을 생성한 다음에 그 자료를 디스켓에 담아 퍼스날 컴퓨터에서 확대문자로 변환하여 인쇄하던가 아니면 계시용의 합격자 명부의 구성항목을 재조정 해야한다.

2.6. 등록금 처리 단계의 실행

이 단계에서 확인가능한 재사용 부품의 종류는 납입금 고지서와 장학생 통지서등의 양식 종류가 있고, 통계표 종류에는 장학생 통계와 등록금 집계표 등이 있고, 각종 명부 종류에는 장학생 명부와 등록생과 미 등록생의 명부가 있다. 등록금 처리는 기본적으로 재무처의 소관업무이나 신입생의 등록금에 관한 한 입시업무와 연관하여 처리 하는 것이 대학 전체의 입장이나 업무 개발담당자의 입장에서 처리하기가 수월하다. 등록금 처리 서브메뉴에 포함되지는 않았으나 등록금 처리 단계에서 미리 해야할 작업들은 파일생성에 관한 작업으로 등록금 파일과 장학생 파일, 고지서 파일등의 작업이 있다. 등록금 처리 단계중 납입금 고지서에 관한 재사용 부품을 확인 하려면 등록금 처리 서브메뉴에서 기능키 F1을 선택한다. 납입금 관련 양식의 포함 항목들은 납입 년도와 학기, 개인의 인적사항으로 학과와 수험번호, 성명등이고, 납입할 항목으로는 입학금과 수업료, 기성회비와 학생회비, 학보 발간비나 표준 교재

대금등의 일괄 납부할 항목이 포함된다. 등록 기간중에는 매일의 납입금 영수증을 수합하고 이를 처리하여 일일 장학생 통계와 등록금 집계표등을 작성하고, 장학 파일과 등록 파일을 갱신하고 등록금 납입기간이 만료되면 등록금 통계표, 장학금 내역표등과 장학생 명부, 등록생과 미 등록생 명부등을 작성한다. 입시관리 주 메뉴에서 등록금 처리 단계로 접근하려면 기능키 F6을 선택한다. 등록금 처리 단계의 서브메뉴는 그림 26과 같다.

```
A.P.R.S.                  등 록 금  처 리  S U B  M E N U          DATE : 94/03/19
입시관리                                                          TIME : 10:10 AM

   COMPONENT ID          FUNCTION KEY          DESCRIPTION

     IPSI610R               F1                  납입금 고지서
     IPSI620U               F2                  일일 등록 처리
     IPSI630S               F3                  장학생 통계표
     IPSI640R               F4                  장학생 연명부
     IPSI650S               F5                  등록금 집계표
     IPSI660R               F6                  등록생 연명부
     IPSI670R               F7                  미 등록생 명부
     IPSI680R               F8                  장학금 내역표
                            F9                RETURN  IPSI MAIN MENU
                            F10               RETURN PRIMOS
                          FUNCTION KEY를 눌러 주시오.

MESSAGE :
Application Program Reuse System                SCREEN-ID : IPSIF600
```

그림 26. 등록금 처리 서브시스템의 메뉴

```
입시관리                  **********                        DATE :  94/03/21
등록처리                  일일 등록금 납입 처리             TIME :  10:10 AM
구분코드 :                **********                        구분:01=광주은행
등록일자 :                                                          02=외환은행

| 순위 | 수험번호 | 성  명 | 납입액 | 순위 | 수험번호 | 성  명 | 납입액 |
|------|----------|--------|--------|------|----------|--------|--------|
|  01  |          |        |        |  02  |          |        |        |
|  03  |          |        |        |  04  |          |        |        |
|  05  |          |        |        |  06  |          |        |        |
|  07  |          |        |        |  08  |          |        |        |
|  09  |          |        |        |  10  |          |        |        |
|  11  |          |        |        |  12  |          |        |        |
|  13  |          |        |        |  14  |          |        |        |
|  15  |          |        |        |  16  |          |        |        |
|  17  |          |        |        |  18  |          |        |        |
|  19  |          |        |        |  20  |          |        |        |

F1 =ENTRY  F2=RESTART  F3=UPDATE  F8=PREMENU  F9=MAINMENU  F10= STOP
MESSAGE :
University. Total. Information. Reuse. System.        SCREEN-ID : IPSIF620
```

그림 27. 일일 등록금 납입 처리 부품

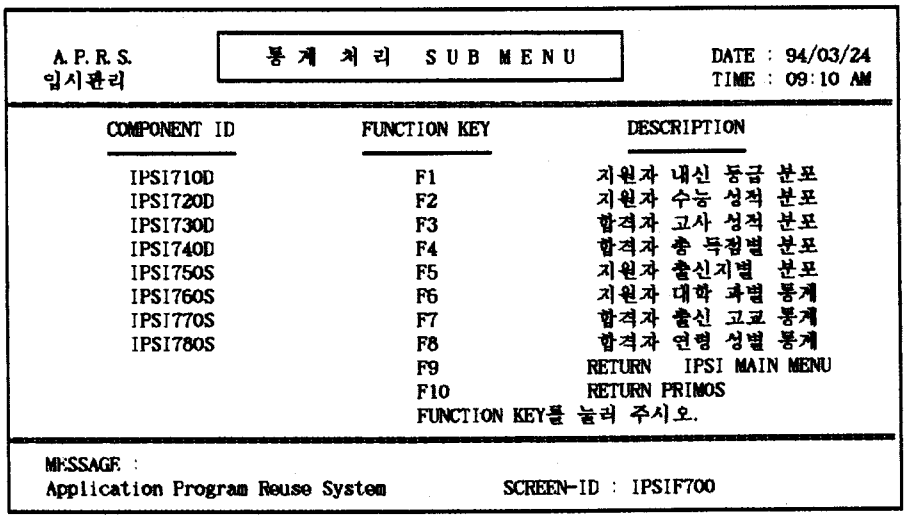


그림 28. 통계 처리 서브시스템의 메뉴

일일 등록금 납입 처리에 관한 재사용 부품을 확인 하려면 등록금 처리 서브메뉴에서 기능키 F2를 선택한다.

그림 27의 일일 등록금 납입 처리 부품에서 알 수 있듯이 납입금 처리는 납입 은행별, 등록일자 별로 한번에 20명 단위로 입력할 수 있다. 부품을 구성하는 항목은 수험번호와 성명, 각 개인의 납입액이며 하루의 입력이 끝나면 처리내용이 갱신된 등록금 마스터 파일에서 각종의 일일 통계를 작성 할 수 있다. 수납처별로 등록상황을 알아보기 위해서는 일일 등록금 납입처리후에 등록금의 일일 집계표를 작성해야 하는데 등록금 집계표에 관한 재사용 부품을 확인 하려면 등록금 처리 서브메뉴에서 기능키 F5를 선택한다. 수납처별 등록금 집계표의 구성항목은 일일 집계항목으로는 수납은행과 각 은행별 납입금액과 납입인원, 금일의 합계금액이 있고 금일 까지의 전체 누계를 집계하기 위한 항목으로는 등록금 누계와 인원 그림 28 통계 처리 서브시스템의 메뉴누계로 되어 있다. 수납 은행명은 각 대학별로 실정에 알맞게 재조정 해야한다.

2.7. 통계 처리 단계의 실행

통계 처리 단계를 실행 하려면 입시관리 주메뉴에서 기능키 F7을 선택한다. 그림 28은 통계 처리 서브시스템의 메뉴이다.

통계 종류에는 지원자와 합격자에 대한 대학별, 학과별 통계와 출신 고교별 통계, 지원자나 합격자에 대한 출신지별 통계와 성별, 연령별 통계등이 있다. 통계 처리 서브시스템에서 확인 가능한 재사용 부품은 지원자나 합격자에 관한 내신 등급 분포표와 수학능력 시험성적 분포표, 지원자나 합격자에 대한 본고사 득점 분포표와 각 고사 성적을 합산한 결과인 총 득점별 분포표 등의 분포도가 있다.

통계처리 서브시스템 메뉴에서 지원자 내신 등급 분포표에 대한 부품내용을 확인 하려면 기능키 F1을 선택한다. 지원자 내신등급 분포표는 대학별로 작성하며, 구성항목은 내신 등급과 과별 소계, 과별로 최고와 최저등급, 평균 등급등이다. 합격자 용으로 쓸 때는 '지원자' 항목을 변경하고 처리 대상 자료는 합격자 마스터 파일을 사용한다.

합격자 수학능력 시험 총점별 분포표의 부품을 확인 하려면 통계처리 서브메뉴에서 기능키 F2를 선택한다. 수학능력 성적 분포표에서 알 수 있듯이 이 부품은 내신 등급과 비슷한 양식을 재조정하여 쓸 수 있으며, 구성 함목에 있어서도 내신등급항목을 수학능력 시험의 총점으로 변경한다.

합격자와 지원자의 출신지별 통계에 관한 부품을 확인 하려면 통계처리 서브메뉴에서 기능키 F5를 선택한다. 이 통계는 각개인의 우편번호에 근거하여 작성하며 구성항목은 지원자와 합격자 출신지별 통계는 대학단위로 작성하며 남녀의 구분은 대각선을 사용하표 1. 성별 연령별 통계여 위에는 남 학생수, 아래는 여학생수를 인쇄한다. 소계는 지역별로 지원자와 합격자를 별개 항목으로 인쇄하고 지원자와 합격자의 총계는 대학별로 산출한다. 합격자 출신고교별 통계는 출신고등학교 코드를 기본 키로 하여 정렬한 다음 인쇄용지 한장에 하나의 고등학교를 작성 할 수도 있으나 이러한 경우 출신고등학교 수 만큼의 인쇄 용지가 소모되며 고등학교간의 대비도 번잡하므로 구현된 재사용 시스템에서는 인쇄 용지 한면에 12개 고등학교를 인쇄하도록 하였다. 출신 고교별 통계의 재사용 부품을 확인하려면 각종 통계처리 서브메뉴에서 기능키 F7을 선택한다. 출신고교별 통계 구성항목은 가로의 행 방향에 각각의 단과 대학명을 쓰고, 세로의 축에는 출신 고등학교 교명을 쓴다. 이 양식은 하나의 양식으로 합격자나 지원자용으로 공용 할 수가 있다. 합격자 연령별 성별 통계는 각 대학별, 학과별로 하나의 인쇄 용지에 출력한다.

입시 관리 / 통계 처리 / 정보 처리 대학 — 94년도 성별 연령별 통계 — DATE : 94/03/30 / PAGE : 118 / (합 격 자)

구분	17이하	18세	19세	20세	21세	22세	23세	24세	25이상	합계
학과	남/녀	남/녀	남/녀	남/녀	남/녀	남/녀	남/녀	남/녀	남/녀	남/녀
	/	/	/	/	/	/	/	/	/	/
	/	/	/	/	/	/	/	/	/	/
	/	/	/	/	/	/	/	/	/	/
	/	/	/	/	/	/	/	/	/	/
	/	/	/	/	/	/	/	/	/	/
대학 계	/	/	/	/	/	/	/	/	/	/

표 1. 성별 연령별 통계

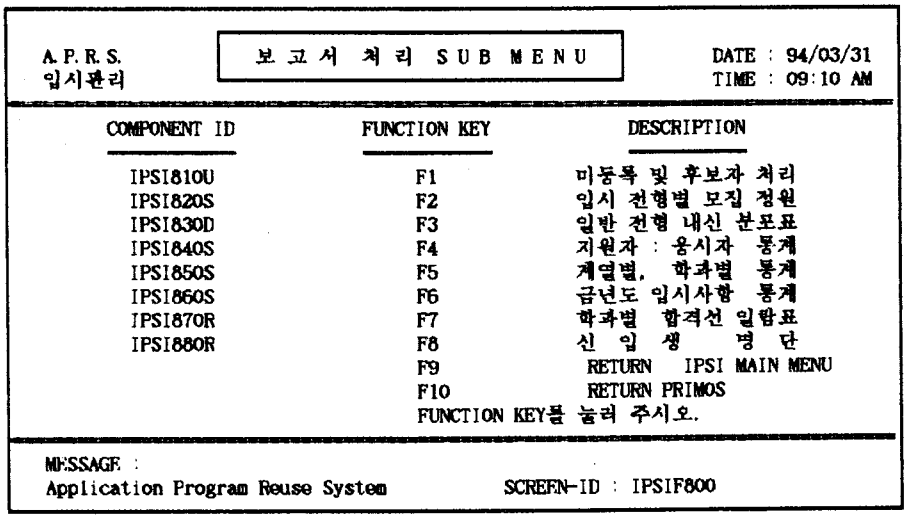


그림 29. 보고서 처리 서브시스템의 메뉴

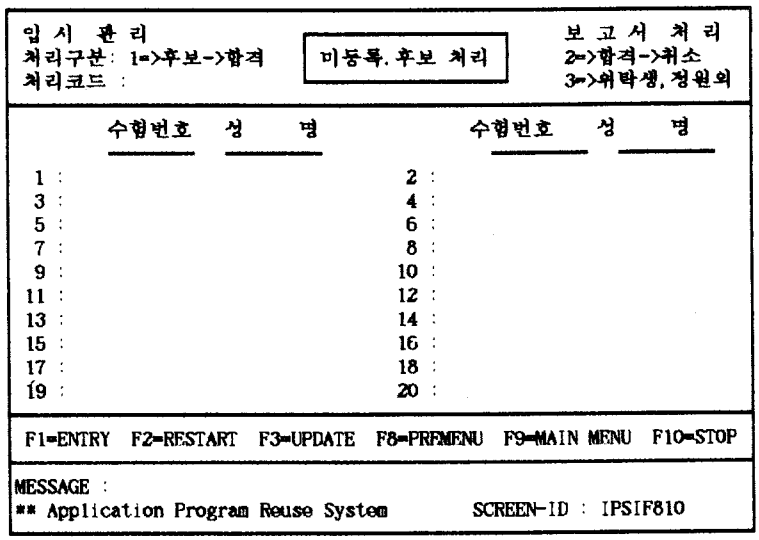


그림 30. 미등록 및 후보자 처리 부품

성별 및 연령별 통계의 재사용 부품을 확인하려면 통계처리 서브메뉴에서 기능키 F8을 선택한다.

표 1은 성별 연령별 통계의 부품 내용으로 하나의 양식으로 지원자나 합격자의 공용으로 사용한다. 부품의 구성항목은 가로의 행에는 연령과 성별을 배열하고 세로에는 해당하는 각 학과의 인원을 배치하며 맨 마지막 줄에 대학계를 인쇄한다.

2.8. 보고서 처리 단계의 실행

보고서 처리 단계를 실행 하려면 입시관리 주 메뉴에서 기능키 F8을 선택한다. 그림 29의 보고서 처리 서브메뉴에서 보는 바와 같이 보고서의 종류에는 그 양식이 교육부에서 하달된 것이 대부분이며 보고서 양식에는 입시 전형별 모집인원, 일반전형의 내신 분포도, 재수생 대비 금년도 졸업생 통계등의 각종 통계와 과별 합격선 일람표, 신입생 명단등이 있다.

보고서 처리 단계에서 가장 먼저 해야 할일은 등록 기간중에 등록금을 내지않은 미등록자를 후보자로 대체하는 작업이다. 미 등록자를 후보자로 대체하는 재사용 부품을 확인 하려면 보고서 처리 서브 메뉴에서 기능키 F1을 선택한다.그림 30은 미등록생 처리 부품으로 구성 항목은 합격 취소인가, 혹은 후보자의 합격자화 인가, 정원외등의 처리인가를 구분하는 처리 코드를 입력하는 항목과 해당되는 각개인의 수험번호를 입력하면 해당되는 수험생의 성명이 화면에 나타난다. 사용자는 이 화면의 성명에 의하여 본인여부를 확인하며 하나의 화면당 20명 단위로 처리가 가능하다. 미등록자와 후보자가 갱신된 완벽한 합격자 파일이 만들어지면 이 파일을 근거로 하여 각종 보고서를 작성 하게 된다.

입시 전형별 모집인원에 관한 통계 양식을 확인 하려면 보고서 처리 서브메뉴에서 기능키 F2를 선택한다.

입시 전형별 모집인원에 관한 부품 양식에는 대학, 학과별로 우선전형인가, 일반전형인가, 주·야간 특별전형, 또는 정원외 특별전형 인가에 따라 해당 인원수를 인쇄한다. 입시 전형별 모집인원은 신입생 모집 요강상의 모집정원과는 다를 수 있으므로 미등록생이 해결된 최종의 확정된 합격자 파일을 근거로 작업이 되어야 한다. 정원외 특별전형이나 위탁생 처리등이 소수의 지원자로 인해 전산처리가 아닌 수작업으로 진행된 경우에는 신입생 파일을 생성하고 학번을 부여한 연후에야 전형별 모집인원은 작업이 가능하다.

지원자 대비 응시자의 통계에 관한 재사용 부품을 확인 하려면 보고서 처리 서브메뉴에서 기능키 F4를 선택한다. 이 통계의 구성 항목은 졸업 년도를 기준하여 금년도 졸업생과 재수생으로 구분하고 각각을 남여 별로 세분한다. 지원자 대비 응시자 통계의 부품내용은 금년 졸업자와 재수생, 지원자와 응시자를 각각 대비하여 남여별로 작성하며 응시율은 지원자/응시자 * 100으로 계산한다. 금년도 입시사항 통계란 계열별 학과별로 지원자 수와 입학생 수를 행으로 배열하고 각 계열과 학과를 열의 요소로 배열하고 마지막 난에 대학 총계를 인쇄하는 2차원의 표이다. 금년도 입시사항 통계의 부품을 확인 하려면 보고서 처리 서브메뉴에서 기능키 F5를 선택한다. 금년도 입시사항 통계의 구성 항목에는 가로축으로 지원자와 신입생 각각에 금년도 졸업자와 재수생, 이들의 합계를 남녀로 구분하여 배치하고 세로축에는 각각의 계열별로 학과를 배치하여 입시사항 통계란 결국은 계열별 학과별 통계가 되는 셈이다.

학과별 합격선 일람표는 교육부 보고사항은 아니며, 대학의 관리자층을 위한 일람표로써

금년도 입시사항 중에서 대학의 전체수석에 관한 사항과 각 대학별, 학과별로 최고 득점자의 인적사항과, 총득점과 본고사 혹은 수능성적등의 최고점수와 최저점수, 합격선 점수등을 대학별로 한눈에 알아보게 작성한다. 신입생 합격선 일람표의 부품을 확인하려면 보고서 서브 메뉴에서 기능키 F7을 선택한다. 부품의 항목중에서 본고사에 관한 최고, 최저 점수항은 본고사를 실시하는 대학의 경우이며, 본고사가 없이 수학능력 시험으로 대치하는 대학은 수능점수로 수정한다. 학과 수석중 최고 득점자가 대학의 수석이 되며, 각 대학의 수석중 최고 득점자가 전체 대학의 수석자가 된다.

교육부 보고용의 신입생 명단에 관한 부품을 확인 하려면 보고서 처리 서브 메뉴에서 기능키 F8을 선택한다. 신입생 명단의 부품 구성항목은 대학 전체 인원에 대한 일련번호와 학과명, 각 개인의 성명을 한글과 한자로 표시하고 주민등록번호, 수학능력 고사에 대한 차수와 수험번호, 총득점, 출신 고교명과 전형구분이다.

V. 결 론

소프트웨어의 재사용은 신뢰성과 생산성이 확보된 기존의 소프트웨어를 재사용 하는것으로 소프트웨어 부품을 그대로 재사용 하던지, 일부의 수정만으로 재사용 한다. 기존 재사용 방법의 문제점은 소프트웨어 수명주기 전 단계중에서 일부분만을 지원하고, 사용자와 개발자간의 시스템 명세 작성시의 착오를 줄일 방안이 없으며, 사용자와 개발자의 두 그룹을 모두 지원하는 방안의 미비, 사용자의 다양한 응용영역의 요구에 미치지 못하고, 사용언어가 특정의 언어이다. 이 문제점을 해결하는 방안으로 본 논문에서는 명세 작성의 정확성을 얻을 수 있는 재사용 부품 확인 및 검색 시스템을 설계하고, 업무영역을 대학업무로 하며, 대학 구성원중 개발자와 사용자가 공용할 수 있는 응용 프로그램 재사용 시스템을 설계하고 대학업무중에서 입시관리 업무를 사용언어는 COBOL로하여 이를 구현하였다.

구현된 응용 프로그램 재사용 시스템은 소프트웨어 수명주기중에서 분석과 설계단계의 산출물인 요구 명세서를 대신하여 그 요구에 대한 컴퓨터의 처리 결과를 검색.확인하게 하여 분석과 설계과정의 오차를 최소화 한다. 만약의 경우에 확인 과정에서 사용자의 변경요구가 있다면, 곧바로 같은 시스템에서 사용자의 수정요구를 처리할 수 있다.

그러나 구현된 시스템은 응용영역이 대학업무의 일부에 국한되어 있고, 구현시 사용한 언어가 COBOL이고 하드웨어도 특정 기종이므로 이에 대한 보완으로 대학업무 전체를 망라한 시스템의 설계와 구현, 프로그래밍 언어와 재사용 시스템의 다변화, 재사용 부품의 생성과 저장 방안등의 보완이 필요하다.

참 고 문 헌

김치수, "추상 자료형을 기반으로 한 컴퍼넌트의 합성과 릴레이션쉽에 관한 연구", 중앙 대학교, 박사학위 논문, 1990. 6.

변상용, "소프트웨어 설계정보의 재사용에 관한 연구", 중앙 대학교, 박사학위 논문, 1990. 12.

이경환외, 소프트웨어 재이용을 위한 연구, 과기처 연구보고서, BSN20592, 1989

전국 대학 전산소장 협의회, 대학전산소 총람, 영진 출판사, 1992.

중앙 대학교, CARS 매뉴얼, 1992.

중앙 대학교, 소프트웨어 재이용에 관한 연구, 과기처 연구보고서, 1993, 10

진영택, "객체지향 시스템에서 클래스 라이브러리 계층의 재구성에 관한 연구", 중앙 대학교 박사학위 논문, 1991.

A Goldberg and D. Robdon, : Smalltalk-80, Addison-wesley, 1989.

A.S Fisher, : CASE : Using software development tools , Wiley, 1988.

B.A.Burton, R.W.Aragon, S.A.Bailey, K.D. Koehler, L.A.Mayer, "The reusable software lablary", IEEE Software, 1987, pp. 25-33.

B. H. Boar, : Application Prototyping, Wiley-Interscience, 1984.

Biggerstaff,t., Perlis,A., "Forward : Special issue on software reusability",IEEE transactions on software enginerring,Vol.SE-10, No.5, Sep. 1984, pp.474-476.

Biggerstaff, t., Richter, c., "Reusability framework,assessment, and directions",IEEE software, Vol.4,No.2, 1987, pp.41-49.

B. Meyer, "Lessons from the designs of the Eiffel labraries", Communications of the ACM/September, Vol.33 NO. 9, 1990, pp. 69-88.

Bruce Barnes, "A framework and economic foundation for software reuse",Software productivity consortium, Reston Va 22091, june 1987, pp. 81-87.

C. Gane, : Computer-Aided Software Engineering, Prentice-Hall International, 1990.

C. Jette and R. Smith, "Examples of reusability in an object-oriented programming environment", Software reusability, Vol. 2, 1989, pp. 73-101.

C.V. Ramamoorthy & V. Carg & A.Prakash, "Support for reusability in Genisis", COMPSAC. 1986, pp.299-305.

C.V. Ramamoorthy , A.Prakasy, W.T.Tasi and Y. Usuda, "Software engineering : Problems perspectives", IEEE Computer, Oct. 1984, pp.191-209.

E.Horowitz and J.B.Munson, "An expansive view of reusable software", IEEE transactions on software enginerring,Vol.SE-10, No.5, Sep. 1984, pp. 477-487.

E.E. Wald, "Software Engineering with reusable parts", Proceeding of COMPCONS'87, 1987, pp. 353-356.

J.M. Neghbors, "The Draco approach to constructing software from reusable components", IEEE transaction on software engineering, Sep. 1984, pp. 564-574.

Jones, T.C, "Reusability in programming : A survey of the state of the art", IEEE transactions on software engineering, Vol. SE-10, No.5, Sep. 1984, pp.488-493.

Jones, G., "Methodology/Environment Suport for Reusability", Proc. of the workshop on SW reusability and maintainability, Oct. 1987.

J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy & W. Lorensen, : Object-oriented modeling and design,Prentice Hall, Inc. 1991.

Kang, K.C, "A reuse-based SW development methodology", Proc. of the workshop on SW reusability and maintainability, Oct. 1987

K.W. Jameson, "A model for the reuse of software design information", Communications of the ACM, , Jan. 1989, pp.205-216.

Lanergan,R.G. and Grasso, C.A, "Software engineering with reusable design and code", IEEE transactions on software engineering, Vol. SE-10, No.5, Sep. 1984, pp.498-501.

L. J. Arthure, B. Kapur, : Measuring programmer productivity and software quality, John Wiley & Sons Inc., 1985.

L.J. Pinson & S. Wiener, : An introduction to Object-Oriented programming and Smalltalk, Addison-Wesley, 1988.

M. A. Cusmano, "The software factory : A historical interpretation", IEEE Software, Mar. 1989.

Matsumoto Yeh, "Organizational effort for reusing existing software", Proc. of COMPCON, Fall 1984.

Matsumoto, Y., "Some experience in promoting reusable software", IEEE transactions on software enginerring, Vol. SE-10, No.5, Sep. 1984,

pp.502-512.

Matsumoto Yeh , "A software factory : An overall approach to software production", Tutorial : Software reusability, Dec. 1986, pp. 155-178.

Meyer, B., "The case for object-oriented programming"., IEEE software,Vol.4,No.2, 1987, pp. 50-63.

M.D. Lubars, "Code reusability in the large versus code reusability in the small", Tutorial : software reuse : Emerging technology, 1988, pp. 33-40.

Mitchell D. Lubars, "Affording higher realiability through software reusability", ACM Sigsoft SE Notes, Vol. 11, No.2, Dct. 1986, pp. 39-42.

M. Lenz, H.A. Schmid, and P.W. Wolf, "Software reuse through building blocks", IEEE Software, Jul. 1987, pp. 34-42.

N. Akima & F. Ooi, "Industrializing software development : A Japanese approach", IEEE Software, Mar., 1989.

Ninamary Buba Maginis, "Specialist : Reusable code helps increase productivity", Computerworld, Nov. 24, 1986, pp. 19-20.

O. Chisio, P. Gouthier, S. Truzzi, "An extended approuch to reusability", Compcon 87 spring, 1987, pp.385-389.

P.A.V. Hall, "Software components and reuse - getting more out of your code", Information and software technology, Feb. 1987, pp. 38-43.

P. Coad and E. Yourdon, : Object-Oriented analysis, Prentice Hall, Inc. 1990.

P. Freeman, "A perspective on reusability, in tutorial : Software reusability", IEEE Computer society press, Nov. 1987

P.H. Loy, "A comparison of object-oriented and structured development methods",Communications of the ACM SIGSOFT, Vol.15, No.1, Jan. 1990, pp. 44-48.

Prieto-diaz,R. and Freeman,P., "Classifying SW for reusability", IEEE software, Jan. 1987, pp. 6-16.

R.G. Lanergan & B.A. Poynton, "Reusable code : The application development technique of the future", Proceeding IBM share/guide software symposium, IBM user group, IBM Corp., Armont, N.K., 1979.

R.S. Wiener & L.J. Pinson, : An introduction to Object-Oriented programming and C++, Addison-Wesley, 1988.

S. Gossain, & D.B. Anderson,. "Designing a class

hierarchy for domain representation and reusability", TOOLS'89, 1989, pp. 201-210.

S.N. Woodfield, D.W. Embley, and D.T. Scott, "Can programmers reuse software?", IEEE software, Jul. 1987, pp. 52-59.

S.P. Arnold, S.L. Stepway, "The reuse system : Cataloging and retrieval of Reusable software", Proceeding of COMPCONS'87, 1987, pp. 376-379.

T.A.Standish, "An Essay on software reuse", IEEE transactions on software enginerring,Vol. SE-10, No.5, Sep. 1984, pp. 494-497.

Tracz, W.J, "Software reuse : Motivators and inhibitors", Proc. of COMPCON S'87, 1987, pp. 358 -363.

UCLA Extension, "The design and analysis of reusable software", lecture note, jun. 1988

V.F.Chen & C.V.Ramamoorthy, "The C Inforamation Abstractors", COMPSAC., 1986, pp. 291-298.

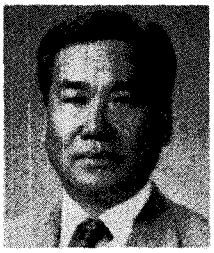
W. B. Rauch-hindin, "reusable software", Electronic design, Feb. 1983, pp. 176-194.

W. Cunningham and K. Beck, "A diagram for Object-Oriented Programs", Proceeding on OOPSLA '86, 1986, pp. 39-43.

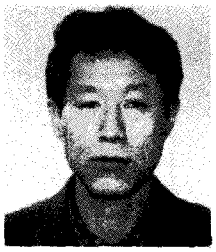
W. Tracz, "Reusability comes of age", IEEE software, Jul. 1987, pp. 6-8.

Yoshinori hori,Shigekatsu kimura, Hiroyuki matsura, "Reusable design methodology for switching SW", Proc. of ISS'87, 1987.

◇ 저자소개 ◇



공동저자 **오무송**은 조선대학교 공과대학 컴퓨터공학과 교수로 재직중이다. 조선대학교 공과대 학 전기공학과를 졸업하고 동 대학원에서 공학석사 학위를 취득하였다. 관심분야는 소프트웨어 엔지니어링과 시스템 소프트웨어 분야이다.



공동저자 **김형태**는 조선대학교 공업전문대학 전자계산과 조교수로 재직중이며 조선대학교 대학 원에서 전자계산 전공으로 이학석사 학위를 취득하고 동 대학원의 박사과정을 이수 하였다. 관심분야는 소프트웨어 엔지니어링과 시스템 분석 및 설계이다.