

컴퓨터 업무상의 기록보전(도큐멘테이션)

김 길 조*
이 무 신**

1. 서 론

컴퓨터 분야에서 시스템 분석과 설계에 관한 기록, 또 프로그램에 대한 설명의 기록을 체계적으로 수행하는 것을 도큐멘테이션(Documentation)이라 한다. 이 도큐멘테이션의 중요성은 흔히 말하면서도, 우리가 조사해 본 바에 의하면 도큐멘테이션이 제대로 되어 있는 곳은 극히 적은 것 같다. 그 근본적인 원인으로서는, 도큐멘테이션의 목적, 구체적 내용 및 작성방식이 충분히 이해되지 못한 때문이라고 본다. 따라서 본논문에서는 대체로 이론적인 내용과 경험을 토대로 도큐멘테이션을 위한 지침을 제시하고자 한다.

2. 도큐멘테이션의 목적

만일 도큐멘테이션이 일차적으로는 시스템 아나리스트나 프로그래머 자기자신을 위해서 하는 것이라는 말에 대하여 의외임을 표시하는 사람이 있다면, 그는 매우 단순한 업무에만 투입되어 온 EDPS 분야 종사자이거나, 무엇인가 업무를 제대로 못하고 있는 자임에 틀림없다. 왜냐하면 매우 복잡하거나, 중요한 업무를 플로우차트(Flowchart) 등의 도큐멘테이션 없이 정확하게 처리한다는 것은 거의 불가능하기 때문이다. 이 점은 업무의 초기 단계에서만 그러한 것이 아니고, 업무가 한참 진행된 뒤나, 심지어는 업무가 완전히 종료한 뒤에도 그러하다. 프로그램의 수

정, 보완이나 제도의 변경에 따른 보충 등 언제나 지나간 일을 검토해 볼 필요는 있으며, 이러한 경우에 "유일한" 기록인 프로그램리스트팅(Program Listing)을 뒤지고 앉아 있는 프로그래머나 심하면 시스템아나리스트를 항상 보는데, 이들은 분명히 업무를 잘못 처리하고 있다. 그들은 정연하게 정리되어 있는 도큐멘테이션을 이용하고 있어야 하는 것이다. 이와 같이, 도큐멘테이션은 무엇보다도 시스템아나리스트와 프로그래머의 업무자체를 위하여라기 보다는 그 업무자체의 내용으로서 요청되는 것이다. 이점에 대한 인식부족이 오늘날 도큐멘테이션의 부재를 초래했다고 할 수 있다.

둘째로, 도큐멘테이션은 자신이 속해 있는 기관, 자신의 상관, 동료, 신입사원, 후임자, 마지막으로 연구지원자(Sponsor)를 위하여 작성하는 것이다.

(1) 컴퓨터 업무는 무형의 두뇌노동인 동시에, 그 경험의 축적이 바로 그 기관의 저력을 형성한다. 따라서 자신의 노동을 유형화함으로써 기관의 경험축적을 가능케 하는 것은 그 기관의 사활과 직결된 문제이다.

(2) 업무가 크면, 작업을 조각내어 여러 사람이 분담개발하는 것이 불가피해지며 이때 동료간의 책임있는 대화(Communication)를 위하여 도큐멘테이션이 불가피함은 명료하다.

(3) 상관은 도큐멘테이션을 통해서 효과적으로 업무의 진행과 내용을 감독할 수 있다.

(4) 새로이 참가한 컴퓨터 요원에 대하여 도큐멘테이션을 공부하게 하는 것보다 더욱 효과적인 교육방법은 없다.

(5) 자기가 퇴직, 출장, 전직, 결근 등으로 그

*인하대학 산업공학과

**KIST 전자체산실

자리에 없게 될 경우 그 프로그램을 이용해야 하는 사람을 위해서, 도큐멘테이션이 유일한 대화 수단으로 남게 된다는 점을 생각하면, 도큐멘테이션이 후임자와 연결되어 가장 자주 거론되는 이유를 알 수 있다.

(6) 연구지원자에게 내는 보고서는 이 도큐멘테이션 이외의 것일 수 없다. 완벽한 보고서는 완벽한 도큐멘테이션을 의미하며, 따라서 컴퓨터 분야의 보고서의 접수기준은 도큐멘테이션이 제대로 편집되어 있는가에 있다. 이 점이 간과되면, 연구지원자 스스로가 개발된 프로그램을 이용하고자 할 때 막대한 곤란에 봉착하게 될 것이다.

3. 플로우 차트(Flowchart)의 종류

도큐멘테이션의 내용을 설명하기 전에, 그 중요부분을 형성할 플로우 차트에 관하여 우선 고찰하기로 한다. 플로우 차트를 사용하는 이유는

(1) 종이위에 그림으로 나타내어진 논리의 흐름은 그 이용자에게 문제된 논리의 내용을 명백히 이해하게 한다.

(2) 정연하게 그려진 플로우 차트는 최상의 대화수법이 된다.

(3) 어떤 주어진 조건아래 발생하는 결과를 플로우차트를 일별함으로써 용이하게 포착할 수 있다. 이런 불가피한 장점 때문에 플로우차트는 업무의 일부로서 그려져야 하며, 도큐멘테이션으로 남겨져야 하는 것이다.

대부분의 컴퓨터분야 종사자들이 플로우차트라 하면, 오직 프로그램을 그대로 나타내는 "프로그램 플로우차트"(Program Specification Flowchart)만을 의미하는 것으로 아는 것 같다. 사실은 플로우차트에는 다음의 6가지가 있으며, 그 어느 하나도 생략할 수 없는 것이다.

3.1 절차 플로우차트(Procedure Flowchart)

절차 플로우차트란 컴퓨터 도입 이전의 구체제에 있어서 정보가 유통되는 절차를 나타내는 그림을 말한다.

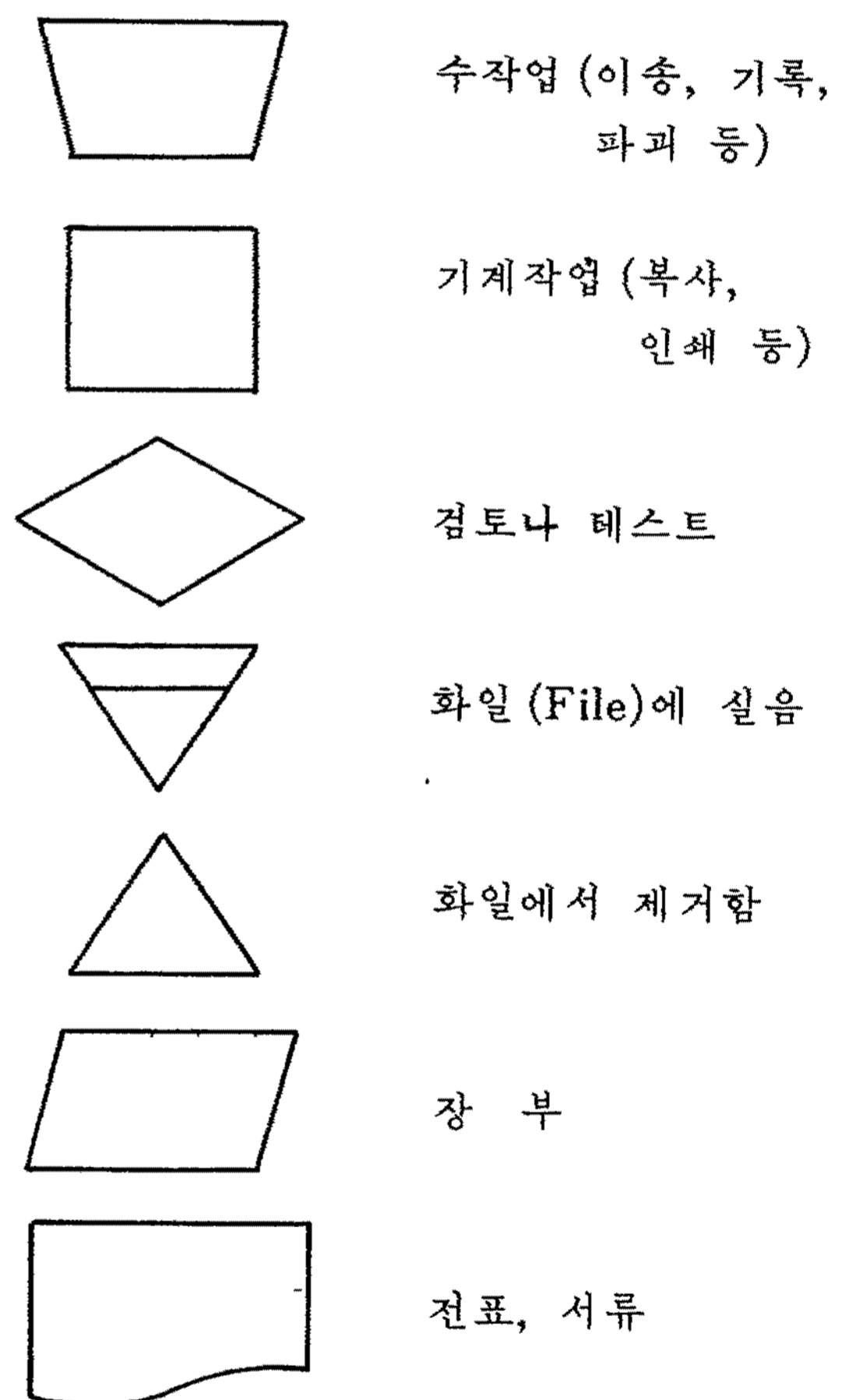
이 플로우차트는

(a) 시스템아널리스트가 분석하는 대상을 명백히 하며,

(b) 그가 분명하고 논리적인 분석을 하는 기초를 제공하며,

(c) 앞으로 새로운 시스템(컴퓨터 시스템)을 이용할 사람들과의 대화의 통로를 확보하게 한다. 이 그림에는 (a) Work Study Symbols (통상, American Standard Method Engineering (A. S. M. E.) Symbol이라 불린다), (b) Horizontal Form Flow Chart Symbols (약해서 H. F. F. C. Symbol이라 한다) 및 (c) Computerbased Symbols (통상 E. C. M. A. Symbol이라 불린다)의 세가지가 있으나, 점차로 E. C. M. A. Symbol로 통일되어 가고 있다.

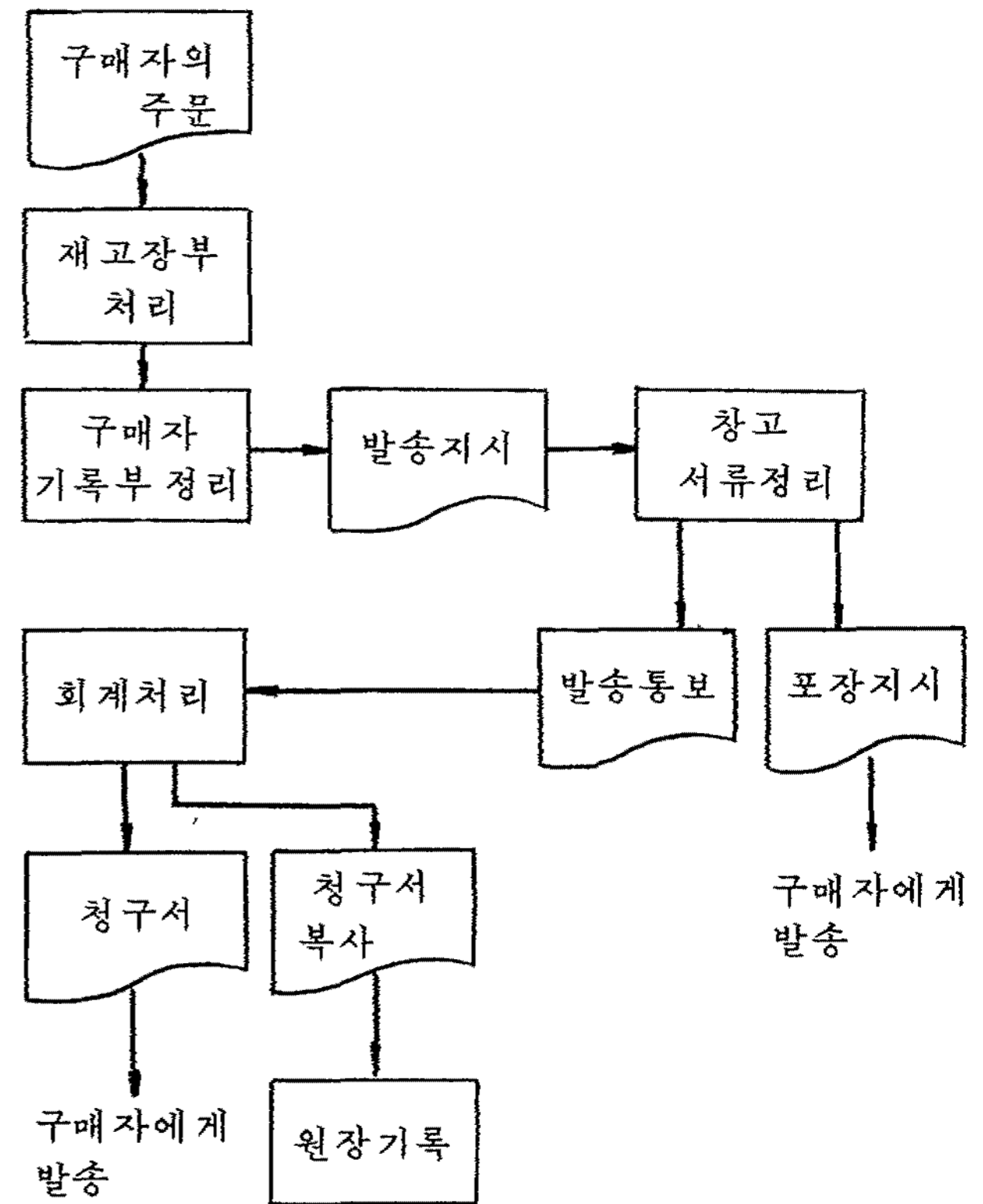
E. C. M. A. Symbol은 다음과 같다.



(A. S. M. E. Symbol이나, H. F. F. C. Symbol에 관하여는 "Flow Chart Symbols," Bureau of Standard, U. S. A. 4058, 1966 참조)

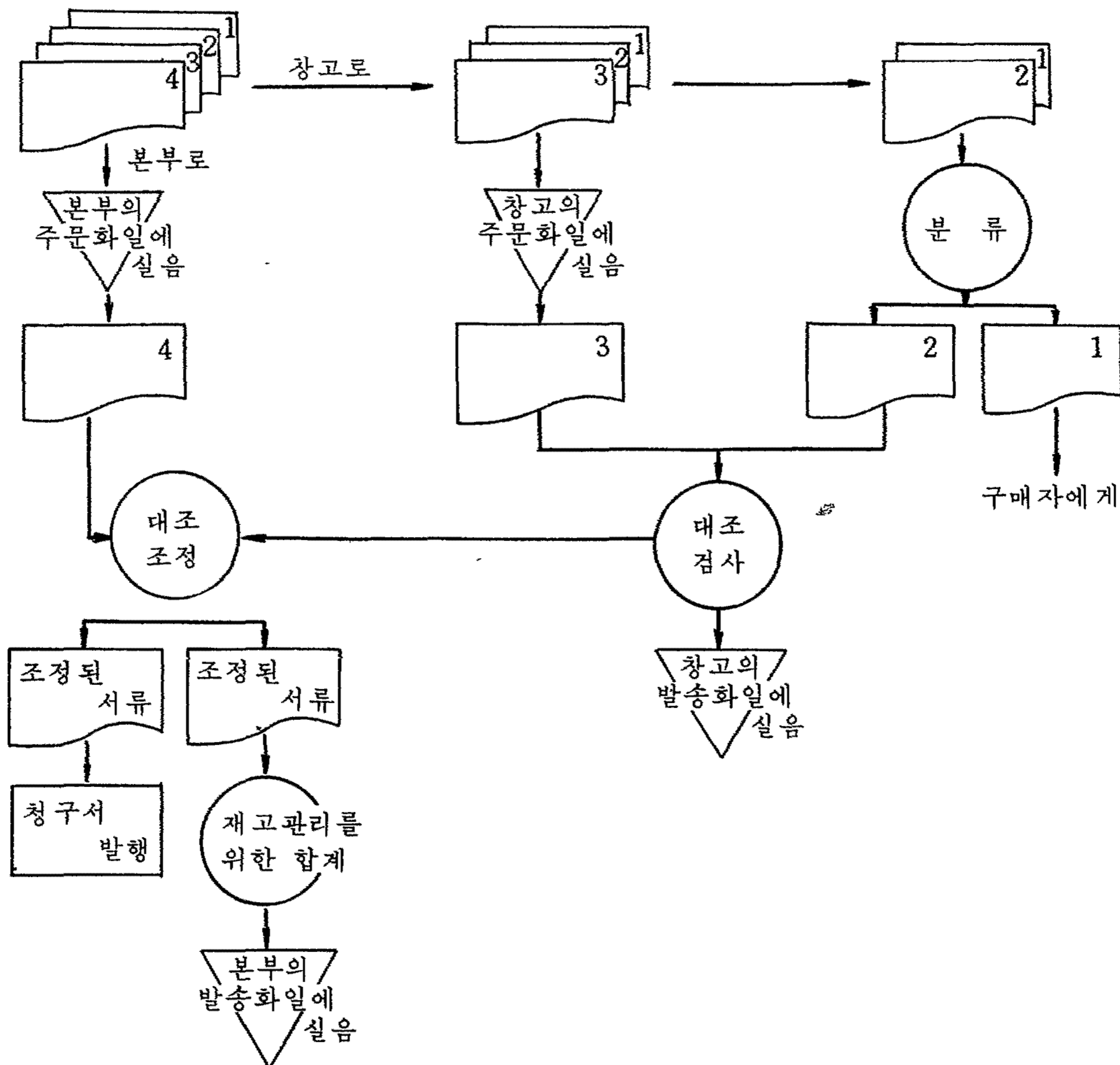
3.2 시스템의 기능별 플로우차트 (System Function Diagram)

위의 절차 플로우차트에 대응하는 새로운 컴퓨터 시스템을 나타내는 그림으로서, 아직 컴퓨터 작업의 도입은 없고, 오직 새로운 시스템에 있어서의 정보의 흐름과 작성되는 출력(Output)만을 나타낸다. 여기서도 E. C. M. A. Symbol이 사용된다. 오른 쪽의 그림은 "시스템의 기능별 플로우차트"의 한 예이다.



3.3 시스템 플로우차트 (System Flowchart)

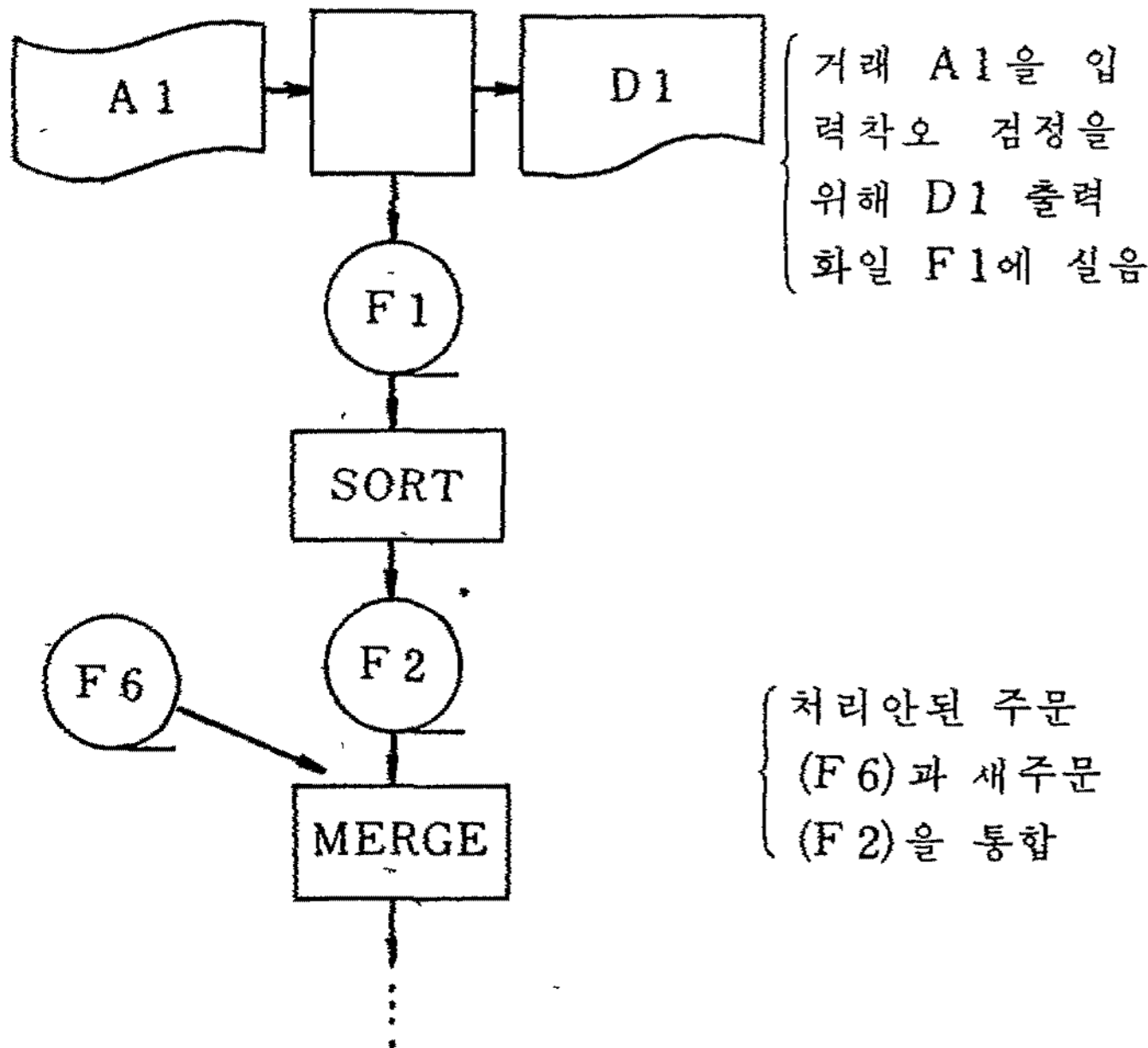
이 그림은 컴퓨터를 중심으로 한 서류의 흐름을 나타내며, 역시 E. C. M. A. Symbol을 이용한다. 그 한 예는 아래와 같다.



3.4 컴퓨터 플로우차트(Computer Flowchart)

이 그림은 위의 시스템 플로우차트의 하나 하나를 구체적으로 설계한다. 즉 이 네모꼴 하나가 입력과 착오의 인쇄, 분류(Sort), 분류된 내용의 입력과 머지(Merge) 등의 작업단위로 쪼개지며, 각 작업단위는 입출력 파일의 명칭이 표시된 플로우차트와 처리되는 업무내용에 대한 설명으로 구성된다. 여기서부터는 통상적인 플로우차트기호를 쓴다.

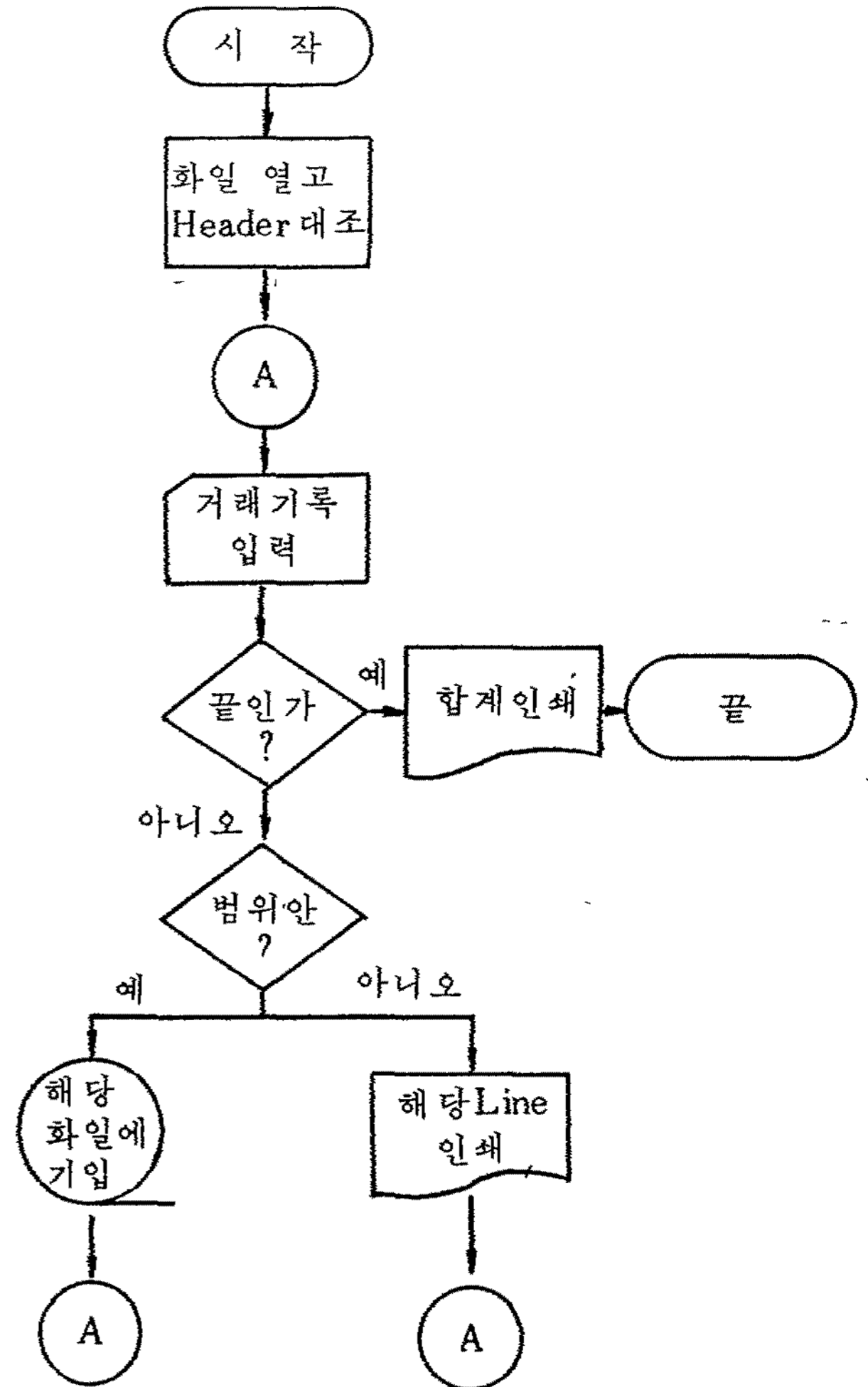
다음의 그림은 컴퓨터 플로우차트의 한 예이다.



3.5 프로그램의 기능별 플로우차트 (Program Function Diagram)

이 그림은 시스템 아나리스트가 작성하는 가장 자세한 플로우차트로서, 프로그래머가 작성하는 프로그램은 이것으로부터 작성된다. 이 그림에는 위의 컴퓨터 플로우차트에 나타나는 작업단위 하나하나가 프로그램별로 나타나며, 필요한 작업이 네모꼴로서 나타난다(예: 파일을 열고, 헤더(Header)를 체크한다; 결과를 프린트한다; 등). 입출력의 자릿수 등을 나타내는 레이아웃(Layout)을 문서화하는 어디에서 찾아쓰라는 지시가 포함된다.

다음은 프로그램의 기능별 플로우차트의 한 예이다.



3.6 프로그램 플로우차트 (Program Specification Flowchart)

프로그래머가 그리는, 프로그램의 내용과 일치하는 플로우차트를 말한다.

4. 플로우차트 작성의 지침

- (1) 논리의 흐름이 명백해야 하므로
 - (a) 표준의 기호를 사용해야 하며,
 - (b) 화살표가 되도록 접치지 않게 하고,
 - (c) 논리의 시작과 끝을 명시해야 하며,
 - (d) 되도록 오른쪽이나 아래 쪽의 일정한 방향으로 논리가 흘러가게 한다.
 - (e) 판단에서 나오는 가지수를 되도록 적게

해야 한다.

(Yes/No; >, =, <; 등과 같이)

- (f) 너무 백백하게 그리는 것을 피하고,
- (g) 되도록 큰 용지를 써서 많은 논리가 한 눈에 들어오게 한다.

- (2) 논리의 직시적인 제시가 목적이므로
 - (a) 여러가지의 가능한 경우가 다 고려되어야 하며,
 - (b) 한 경우가 되풀이 고려되어서도 안되고,
 - (c) 논리의 흐름 전체가 처음부터 끝까지 일사불란하게 제시되어야 하며,
 - (c) 위의 여섯가지 플로우차트의 수준 (Level)이 혼동되어서는 안된다.
- (3) 제대로 작성되었는지 반드시 검증하여야 하며, 이는 단순한 테스트자료를 대입해 봄으로써 알 수 있다.
- (4) 플로우차트는 일일이 상급자에 의하여 검토, 확인되어야 하며, 상급자의 서명으로써 이 행위가 기록으로 남겨져야 한다.

5. 도큐멘테이션의 내용

시스템아나리스트는 다음과 같은 도큐멘테이션을 작성완료하고, 이것은 프로그래머에게 보내진다. 이 도큐멘테이션에 의하여 프로그래머의 작업은 이루어지며, 프로그래머가 완성된 프로그램 리스팅(Program Listing)과 프로그램 플로우차트(Program Flowchart)를 이에 첨가함으로써 도큐멘테이션은 완료된다.

시스템아나리스트가 작성하는 도큐멘테이션

제 1장 서론

- 1. 처리할 일의 제목과 책임소재
 - (1) 시스템 이름
 - (2) 시스템 번호
 - (3) 작성일자
 - (4) 작성자의 이름, 주소, 전화번호
 - (5) 확인자의 이름, 주소, 전화번호
 - (6) 연구지원자의 이름, 주소, 전화번호
 - (7) 본 도큐멘테이션의 수정을 명할 수 있는 부서명

- 2. 책임의 부담
 - (1) 책임부담을 나타내는 문구
 - (2) 연구책임자의 서명
 - (3) 책임한계의 명시
- 3. 도큐멘테이션의 목차
- 4. 수정에 관한 기록
 - (1) 수정번호
 - (2) 수정을 하거나 명한 부서명
 - (3) 수정일자
 - (4) 수정부분과 수정내용

제 2장 시스템의 분석 및 설계

- 1. 절차 플로우차트 (Procedure Flowchart)
- 2. 관련된 업무부서의 명시
- 3. 분석되는 시스템이 전체 시스템의 일부인 경우에는, 관련된 절차를 명시해 놓는다.
- 4. 시스템 분석의 이익
 - (1) 금전적 이익
 - (2) 인력 운영상의 이익
 - (3) 업무 관리상의 이익
 - (4) 정보 이용상의 이익
 - (5) 정보의 질향상에 미치는 영향
- 5. 출력 (Output)
 - (1) 출력을 담을 시설의 지정
 - (2) 출력의 내용에 대한 설명
 - (3) 그 출력에 쓰이는 방법
 - (4) 그 출력이 나타나는 "프로그램 기능별 플로우차트"의 해당부분의 명시
 - (5) 출력의 양
 - (6) 출력의 빈도와 작업 관련사항
- 6. 입력 (Input)
 - 출력과 같은 내용
- 7. 화 일 (File)
 - 출력과 같은 내용

제 3장 구체적 시스템 기술 (Description)

- 1. "시스템 기능별 플로우차트" (System Function Diagram)
 - (1) 플로우차트
 - (2) 다음 사항에 관한 설명

- (a) 일/주/월별 조치
- (b) 코드체제
- (c) 관리체제
- (d) 자료의 수송 및 취급 체제
- 2. "시스템 플로우차트" (System Flowchart) 플로우차트
- 3. "컴퓨터 플로우차트" (Computer Flowchart)
 - (1) 플로우차트
 - (2) 다음 설명에 관한 설명
 - (a) 일/주/월별 프로그램 사용회수
 - (b) 프로그램 사용의 목적과 기능

제 4장 업무전환절차

- 1. 화일의 전환
 - (1) 컴퓨터 시스템을 위한 자료수집방법에 관한 설명
 - (2) 필요한 편집내용에 관한 설명
 - (3) 각종 코드
 - (4) 자료의 정리에 관한 사항
 - (5) 화일전환에 필요한 프로그램의 명시
 - (6) 구체적으로 프로그램사용의 순서를 명시
- 2. 예비운영 (Pilot Running)
 - (1) 이용될 자료
 - (2) 처리될 자료의 양
 - (3) 운영시간 또는 기간
 - (4) 기대되는 결과
 - (5) 앞으로 제기될 관련사항
- 3. 병행운영 (Parallel Running)
 - (1) 운영기간
 - (2) 처리해야 하는 자료의 양
 - (3) 점검시점
- 4. 전환업무 자체
 - (1) 전환기간
 - (2) 초과근무 (Overtime Work)에 대한 추정
 - (3) 추가적으로 고용되어야 할 근무자에 대한 계획
 - (4) 기대되는 업무의 지연

제 5장 시스템 설계의 전제가 된 기재에 관한 설명

- 1. 기재의 지정
 - (1) 컴퓨터의 종류
 - (2) 주변기기
 - (3) 보조장비
- 2. 컴퓨터의 이용
 - (1) 업무단위 (Run)의 수
 - (2) Run의 회수
 - (3) Run에 소요되는 준비시간 (Job Card의 준비 등)
 - (4) Run에 소요되는 시간 (최소, 최대 및 평균시간)
 - (5) 일/주/월별로 소요되는 Run시간의 합계
 - (6) 주기에 따른 피크타임(Peak Time)의 고려
- 3. 주변기기 및 보조장비의 이용
 - 위의 컴퓨터의 이용과 같은 내용

제 6장 입력

- 1. 입력내용
 - (1) 이름, 번호, 목적 등
 - (2) 어떻게, 어디서 작성되는가?
 - (3) 자료의 각 요소 (Elements of Data)
 - (a) 각 요소의 명칭
 - (b) 각 요소의 길이
 - (c) 가능한 최대값
 - (d) 각 요소의 출처
 - (4) 빈 도
- 2. 입력내용의 기본
 - (1) 입력 매체에 따른 Sample
 - (2) 실제 이용되는 현상자료 Sample

제 7장 출력

- 1. 출력 레이아웃 (Layout)
 - (1) 출력의 이름 및 번호
 - (2) 관련된 시스템 이름 및 번호
 - (3) 관련된 프로그램의 이름 및 번호
 - (4) 각 페이지에 인쇄될 행 (Line)의 수
 - (5) 가장 긴 행 (Line)의 길이
 - (6) 행 (Line)에 있어서의 띄어쓰기에 관한

지시

(7) 행 사이의 띄어 두기에 관한 지시

2. 위의 레이아웃에 관한 설명

(1) 목 적

(2) 사전 인쇄된 양식을 양식을 쓰는 경우에는 그 크기, 공백란, 저질, 사전인쇄 내용 등

(3) 기타 특수한 사항 (예 : 제한된 자체의 이용)

(4) 출력의 배포 및 통제에 관한 사항

3. 표본 첨부

제 8 장 화 일

1. 모든 화일에 대하여,

(1) 주변장치명

(2) 이 름

(3) 레이블 (Label)

(4) 길이 (최대, 최소 및 평균)

(5) 레코드 (Record) 종류의 수

(6) 일련번호

(7) 블럭 (Block)의 크기 (Unit of Transfer)

(8) 레코드 (Record) 안의 각 필드에 관한 설명

(a) 이 름

(b) 설 명

(c) 길 이

(d) 최대값

(e) 허용되는 값 (예 : 음수, 양수 등)

2. 특히 카드화일은

(1) 카드타입

(2) 필드당 칼럼 (Card Column)의 수

(3) 천공상의 주의사항

3. 페이퍼 테이프 (Paper Tape) 화일

(1) 블럭사이의 거리 (Inter-block Gap)

(2) 블럭의 시작과 종료기호 및 필드 표시기호 (Field Marker)

(3) 천공상의 주의사항

4. 자기 테이프 (Magnetic Tape) 화일

(1) 암호

(2) 기록밀도 (Density)

(3) 릴 (Reel)의 수

(4) 사용빈도

(5) 다시 사용할 때까지의 기간

5. 매스 스토리지 (Mass Storage) 화일

(1) 저장방식 (Random, Sequential 등)

(2) 접근방식 (Address Computation 등)

(3) 버킷 (Bucket)의 크기

(4) 버킷의 수

제 9 장 시스템의 테스트를 위한 자료

1. 입력자료

리스팅 (Listing) 과 레이아웃 (Layout)

2. 화 일

레이아웃

3. 기대되는 결과

(1) 논리적 또는 산술적 해당

(2) 라인 프린터의 결과 (제 7 장의 출력 레이아웃)

제 10 장 프로그램 기술

1. 서 론

(1) 프로그램 명칭

(2) 프로그램 번호

(3) 프로그램의 목적

(4) "프로그램 플로우차트"

2. 시작 절차

비표준형의 시작 절차를 쓸 때는 그에 대한 설명

3. 본 절차

(1) 처리조건

(2) Validity Check에 대한 설명

(3) 결과확인을 위한 합계에 관한 설명

(4) 최소 및 최대크기 (Size)에 관한 지적

(5) Sign에 관한 지적

(6) 반올림에 대한 조치에 대한 설명

(7) Error가 발생할 수 있는 조건에 대한 설명

(8) Batch Control Program 설명

(9) Batch Control 데이터의 설명

(10) Switching 관계의 지시

4. 마지막 절차

(1) 결과확인을 위한 합계, 도표에 관한 설명

- (2) Dump할 경우에 대한 지적
- (3) 화일의 닫음 (Closing)에 관하여 특기사항이 있으면 기술
- (4) Operator에게 가는 메시지
- (5) 어느 프로그램에 연결 (입력)되는가에 대한 설명

5. Dump/Restart 절차

- (1) 작업중단시기에 관한 설명
- (2) Dump할 주변기기의 지정
- (3) Dump할 내용에 대한 지정

6. 결 어

이상에서 도큐멘테이션에 관하여 설명하였다. 특히 주의를 요하는 것들은 다음과 같이 요약할 수 있다.

- (1) 도큐멘테이션은 업무의 일부로서 수행되는 것이 아니라 업무가 끝난 후에 정리작업으로서 하는 것은 아니다.

(2) 도큐멘테이션의 계층성 (Levels)

도큐멘테이션에는 우선 시스템 아날리스트가 하는 도큐멘테이션이 있고, 프로그래머가 하는 도큐멘테이션이 있다. 다시 시스템 아날리스트가 하는 도큐멘테이션에는 (a) 절차 (b) 시스템의 기능 (c) 시스템의 분석 (d) 컴퓨터 도입 및 (e) 프로그램 기능의 5가지 계층이 있으며, 어느 하나도 생략되거나 합병될 수 없다.

(3) 표준화 (Standardization)의 필요성

도큐멘테이션은 대화 (Communication)의 확립과 신속화를 목적으로 하므로, 그 양식,

내용과 기호 등의 표준화가 필요하다. 이 중 기호에 대하여서는 미국표준국 (Bureau of Standard)에서 제정한 것을 사용하는 것이 무난할 것이다.

(4) 자신을 위해서 한다.

도큐멘테이션은 일차적으로 자신을 위해서 한다. 이의 이해가 도큐멘테이션 부재를 일소하는 시발점이 된다. 또한 도큐멘테이션은 실제로 이루어지고 있다. 왜냐하면 도큐멘테이션 없이는 업무를 제대로 수행할 수 없기 때문이다. 이렇게 실제로 이루어지는 도큐멘테이션을 계층에 맞추어 정연하게, 명시적으로 확립하는 것이 필요하다.

(5) 상급자의 감독의무

도큐멘테이션이 궁극적으로는 그 소속기관을 위한 것이라면, 그것이 제대로 이루어지는지를 감독할 의무는 중요한 것이라 하겠다. 특히 플로우차트를 검토하고 연서하는 절차는 엄격히 수행되어야 할 것이다.

참 고 문 헌

- (1) Bureau of Standard, "Flow Chart Symbols," B. S. 4058, 1966, N. Y.
- (2) Kuehne, R. S., H. W. Lindberg, and W. F. Baron, *Handbook of Computer Documentation Standards*, Prentice-Hall Inc. Englewood Cliffs, N. J. 1973
- (3) 中央大學校文獻研究室, ISO RECOMMENDATION FOR DOCUMENTATION, 第1輯 1973.9.