시스템 분석 설계

Chapter 04
자료흐름도

목차

학습목표

- 구조적 분석 방법론의 모형화 도구인 자료흐름도의 특징을 이해한다.
- 자료흐름도의 구성요소인 네 개의 심볼을 식별하여 작성할 수 있도록 학습한다.
- 자료흐름도의 작성 과정에서 놓치기 쉬운 작성 원칙들을 살펴본 후 적용한다.
- 자료흐름도의 작성 사례를 평가하고 개선할 수 있도록 다양한 사례를 검토한다.

01. 요구사항

■ 요구사항

▪ 요구사항의 정의

- 사전적 정의: '이용자가 어떤 문제를 풀거나 목표를 달성하는 데 필요한 조건이나 능력'
- 소프트웨어 개발에서의 정의: '사용자와 개발자가 합의한 범위 내에서 사용자가 필요로 하는 기능'
  시스템이 제공하는 기능 요구와 품질과 같은 비기능 요구로 나뉨
- 요구사항이 정확히 무엇인지 파악하는 작업은 요구분석 단계에서 이루어짐

그림 4-1 사용자의 요구사항을 파악하는 생산자

02. 요구분석의 이해

■ 요구분석의 정의와 목적

- 집짓는 과정의 예시
  - 집을 지으려는 사람은 본인이 수집한 자료와 구상한 내용을 토대로 건축 설계사에게 어떤 집을 짓고 싶은지 설명
  - 본인이 가지고 있는 예산과 언제 입주하고 싶은지, 건축에 필요한 제반 사항 등에 대해서도 이야기를 나눔
  - 이 과정에서 고객은 어떤 건물을 원하는지 충분히 설명할 수 있고 건축 설계사는 고객이 원하는 것을 설계 도면에 빠짐없이 반영할 수 있음

그림 4-2 건축 설계 과정

02. 요구분석의 이해

■ 요구분석의 정의와 목적

- 요구분석의 정의
  - 컴퓨터 용어사전: '시스템이나 소프트웨어의 요구사항을 정의하기 위해 사용자 요구사항을 조사하고 확인하는 과정
  - 요구분석은 소프트웨어 개발 성패의 열쇠
- 요구분석의 목적
  - 목적사용자에게서 필요한 요구사항을 추출해 목표하는 시스템의 모델을 만들고 '요구분석명세서'를 작성하기 위해서
  - 요구분석명세서
    - » 요구분석 단계에서 생성 되는 최종 산출물로 시스템의 기능이 무엇인지(what)에만 초점을 두고 정리
    - » 요구분석단계 후 설계 단계에서는 '설계서'가 만들어지는데 이 문서는 어떻게(how) 구현할지 기술

그림 4-3 '무엇'과 '어떻게'

02. 요구분석의 이해

■ 요구분석의 정의와 목적

- 요구분석 관련자(대학 학사관리시스템 개발의 예시)
  - 발주사: 외주 방식으로 학사관리시스템 개발을 의뢰한 A대학교
  - 경영자(총 책임자): 학사관리시스템 개발을 결정한 A대학교 총장
  - 발주 담당자: 학사관리시스템 개발을 외주로 진행하는 모든 절차를 준비하는 담당자
  - 사용자: 외주 방식으로 개발된 학사관리시스템을 실제 사용하는 사람 (학생, 교수, 조교, 학사담당직원 등)
  - 수주사: A대학교의 학사관리시스템 개발을 위한 응찰 경쟁에서 이겨 개발이 결정된 업체 (B개발사)
  - 분석가: 사용자 요구사항이 무엇인지 파악하고 추출 및 정리하는 것까지 담당
    B개발사의 분석가는 A대학교를 방문해 현행 시스템 의 현황과 문제점을 파악하고 새로운 요구사항을 수집해
    최종 산출물인 요구분석명세서를 작성
  - 설계자: 요구분석명세서를 바탕으로 아키텍처 설계, 모듈 설계, 데이터베이스DB 설계, 사용자 인터페이스 설계 등을 담당
  - 개발자: 넓은 의미에서 개발자는 학사관리시스템 개발에 참여하는 B개발사의 분석가, 설계자, 프로그래머 등을 포함
    좁은 의미에서는 프로그래머

02. 요구분석의 이해

■ 요구분석의 어려움

- 문제 영역에 대한 분석가의 이해력 부족
  - 법적인 문제를 해결할 경우 해당 분야에 전문적인 변호사를 선임
  - 요구분석가는 IP 분야의 전문가라 문제 영역(회계 분야, 금융 분야, 기업의 통합관리 등)에 대한 이해력이 부족할 수 있음
  - 사용자 요구를 잘못 이해한 채 분석해 필수 요구사항을 빠뜨릴 수도 있음
  - 이런 문제를 최소화하려면 문제 영역과 관련된 프로젝트를 개발한 경험이 많은 분석가를 투입하는 것이 바람직

그림 4-4 분야별 전문 변호사의 활동 모습

그림 4-5 경험 많은 분석가들

02. 요구분석의 이해

■ 요구분석의 어려움

- 사용자와 분석가의 의사소통 문제
  - 소프트웨어 개발에서는 견본이 없는 경우가 대부분이므로 사용자가 분석가에게 요구 사항을 설명하기가 어려움
  - 원하는 바를 분석가에게 어떻게 설명해야 할지 잘 모름
  - 필요한 요구사항은 반영하지 못하고 불필요한 사항만 포함되기도 함
  - 의사소통 문제는 개발자 간에도 있을 수 있음
- 사용자의 계속되는 요구사항 추가
  - 사용자의 처음 요구는 단순하고 간단할 수 있지만 분석가와 대화 후 점점 새로운 생각이 늘어나거나 관련 지식이 늘어남
  - 그에 따라 새로운 요구가 생기고 요구사항이 계속 추가될 수 있음
  - 소프트웨어 개발 과정 중에 목표 환경이 달라지면서 요구사항이 변경되기도 함

그림 4-6 요구사항을 계속 추가하는 사용자

02. 요구분석의 이해

■ 요구분석의 어려움

- 사용자의 모호한 요구사항
  - 사용자는 분석가에게 모호하게 요구사항을 전달하면 분석가가 해석할 수 있는 여지가 많으므로 오해로 인한 문제가 발생
  - 세부적인 내용을 전달하지 않는다면 분석가는 세부적인 요구사항을 본인의 경험에 비추어 해석해 사용자의 의도와 달라짐
  - 부서 간의 요구가 서로 어긋나고 경영진, 실무자의 요구가 상반되어 일관성이 없고 모순되는 요구사항이 발견될 수 있음
  - 분석가는 이러한 요구를 정리해서 반영 하기 위해 이해 당사자 간의 주장을 조율해야 함

그림 4-7 조율에 능숙해야 하는 분석가

02. 요구분석의 이해

■ 요구사항 수집

▪ 자료 수집

- 가장 먼저 할 일은 A대학교의 현행 시스템에서 모든 업무의 입력 화면과 출력물을 받아 기본 기능을 분석하는 것
- 관련 직원으로부터 현행 시스템의 문제점과 개선점, 새로 추가되어야 할 요구사항을 파악

▪ 인터뷰

- 가능하면 먼저 자료를 수집한 후 이를 정리해 분석한 결과를 바 탕으로 각 부서 담당자를 인터뷰하는 것이 효율적
- 미리 받은 자료를 토대로 대화를 해야 큰 줄기를 흩트리지 않고 계획대로 진행할 수 있음
- 오랜 시간 동안 대화하기보다는 핵심 요구사항을 빨리 습득하는 것도 능력

▪ 설문 조사

- 설문 조사를 통해 또 한 번의 요구사항을 도출할 수 있음
- 중요한 것은 효율적인 설문 조사를 위해 설문 문항을 잘 만들어야 한다는 것

03. 요구분석 절차와 요구사항 종류

■ 요구분석 절차와 요구사항 분류

▪ 요구 분석 절차

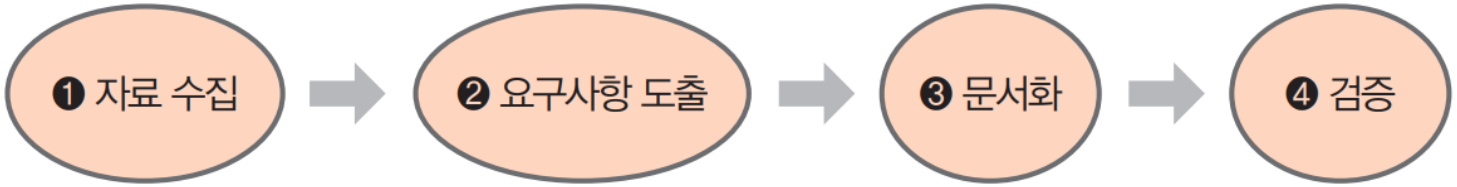

그림 4-8 요구분석 절차

❶ 자료 수집

현행 시스템의 문제점 도출, 실무 담당자 인터뷰, 현재 사용하는 문서 검토 등을 통해 가능한 모든 자료를 수집

❷ 요구사항 도출

수집한 자료를 정리해 적절히 분류하고 개발에 반영할 요구사항을 도출

❸ 문서화

도출한 요구사항을 요구분석명세서로 작성

❹ 검증

요구분석명세서에 사용자 요구가 정확히 기록되어 서로 모순되는 사항은 없는지, 빠뜨리지 않고 전부 기록했는지 등을 점검

03. 요구분석 절차와 요구사항 종류

■ 요구분석 절차와 요구사항 분류

▪ 요구 분석 분류

– 사용자 요구사항은 개발될 소프트웨어가 제공할 기능 요구사항과 성능, 제약 사항, 품질과 같은 비기능 요구사항으로 분류

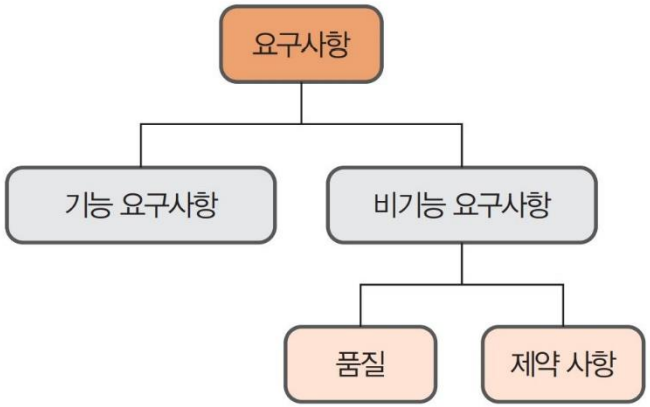

그림 4-9 요구사항 분류

03. 요구분석 절차와 요구사항 종류

■ 기능 요구사항

- 사용자가 원하는 기능
- 사용자는 시스템을 통해 기능을 제공받기 바라며 시스템은 사용자에게 필요한 기능을 제공
- 사용자가 원하는 기능은 요구분석명세서에 완전하고 일관성 있게 표현해야 하며 시스템에도 전부 반영
- 완전성: 사용자가 원하는 모든 기능이 포함 / 일관성: 요구사항 간에 모순이 있어서는 안 됨

■ 비기능 요구사항

• 비기능 요구사항의 개요

- 수행 가능한 환경, 품질, 제약 사항 등을 말함
- 비기능 요구사항이 매우 중요한 소프트웨어에는 군사 무기나 의료 장비 등이 해당

▪ 제약 사항

- 개발 소프트웨어가 수행될 환경에 의한 조건
- 예시
  - » 자바 언어를 사용해 개발하고, CBD 개발 방법론을 적용해야 함
  - » 레드햇 리눅스 엔터프라이즈 버전에서 실행해야 함
  - » 웹 로직 서버를 미들웨어로 사용해야 함
  - » 윈도 운영체제와 리눅스 운영체제에서 모두 실행할 수 있어야 함

03. 요구분석 절차와 요구사항 종류

■ 비기능 요구사항

▪ 기능

• 신뢰성

- 소프트웨어에서의 신뢰성: 소프트웨어를 믿고 사용할 수 있는 것
- 일반적으로 신뢰성은 신뢰도로 나타내며 신뢰도는 '장애 없이 동작하는 시간의 비율'로 나타냄
- 소프트웨어를 1,000번 수행했을 때 오류 없이 동작하는 횟수가 999번인 경우: 99.9%
- 1,000시간 작동하는 동안 1시간의 장애가 있었던 경우: 99.9%(신뢰도 측정은 가용성(이용 가능성)을 척도로 사용)

- 신뢰도 높은 소프트웨어의 조건
  - » 시스템에 오류가 있더라도 고장을 일으키지 않고 회피할 수 있는 능력이 높아야 함
  - » 고장이 발생하더라도 이전 수준으로 성능이 회복되어야 함
  - » 고장으로 영향을 받은 데이터들이 정상적으로 잘 복구됨

03. 요구분석 절차와 요구사항 종류

■ 비기능 요구사항

▪ 기능

• 가용성

– 소프트웨어가 총 운용 시간 동안 얼마나 정상적으로 고장 없이 가동되었는지를 비율로 나타냄

가용성(%) = (MTBF/(MTBF + MTTR)) × 100(%) = 999/(999 + 1) × 100(%) = 99.9%

그림 4-10 신뢰도 계산

– MTBF(평균 정상 작동 시간 또는 고장 간 평균 시간)

» 고장이 나서 수리 후 정상 작동 하다가 다시 고장이 날 때까지 작동한 시간
» 정상 작동 시간의 합을 횟수로 나눈 값
» MTBF와 비교되는 개념으로 고장까지의 평균 시간 (고장은 수리가 불가능한 상황에 해당)

– MTTR(평균 수리 시간)

» 수리 시간의 합을 횟수로 나눈 값
» 수리 시간: 고장 시점에서 수리가 완료되어 재가동되는 시점까지의 시간

04. 요구사항의 표현

■ 표현과 모델의 이해

- 표현
  - 생각하거나 느낀 것을 표현하는 방법은 다양
  - 수학 공식은 글로 표현하면 너무 길고 불분명해지는 내용을 공통으로 이해할 수 있고 오해의 소지가 없는 특정 기호로 표현
- 모델
  - 표현을 위해 사용하는 악보나 수학 공식은 하나의 모델
  - 장난감 자동차나 로봇, 생명 과학자들이 사용하는 DNA 분자 모형 등 또한 모델의 종류

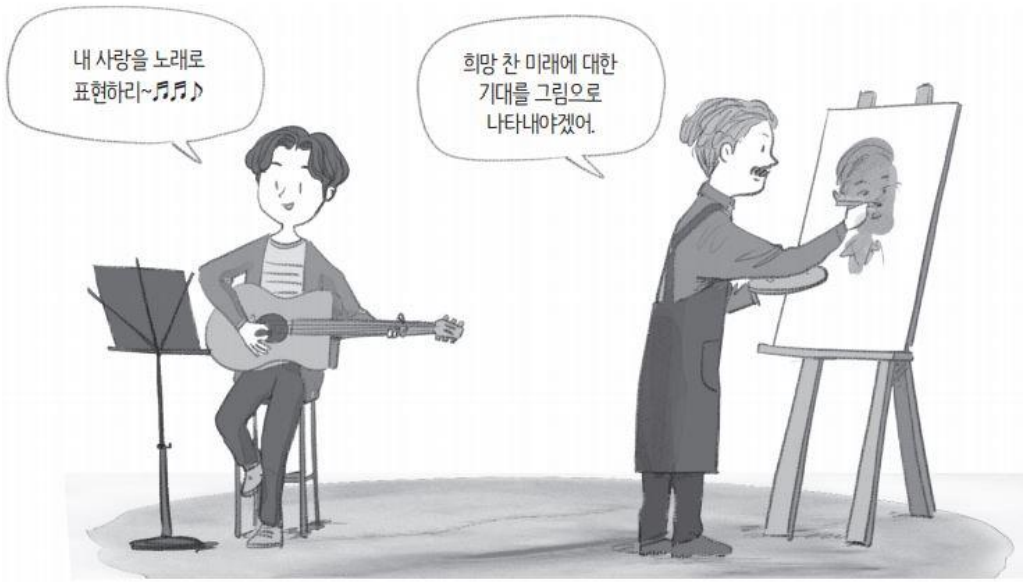

그림 4-11 다양한 표현 방법

04. 요구사항의 표현

■ 모델의 정의와 필요성

▪ 모델의 정의와 필요성

- '복잡한 대상의 핵심 특징만 선별해 일정한 관점으로 단순화한 뒤 기호나 그림 등을 사용해 체계적으로 표현한 것
- 모델이 필요한 가장 큰 이유는 실제 모습을 미리 확인하기 위함(예: 모델하우스)
- 하나의 사물도 여러 모델이 필요(예: 건물 건설시 여러 관점과 사용 목적에 따라 다수의 도면이 필요)

▪ 소프트웨어 개발 모델

- 건축에 여러 도면이 필요하듯이 소프트웨어를 개발할 때도 여러 종류의 모델이 필요
- 객체지향 개발에서는 UML의 다양한 다이어그램을 통해 개발 소프트웨어의 범위나 개략적인 구조와 기능을 이해할 수 있음
- UML의 수많은 다이어그램이 소프트웨어 개발 과정에서 하나의 모델로 사용

04. 요구사항의 표현

■ 모델의 정의와 필요성

▪ 소프트웨어 개발 모델

- 소프트웨어 개발 모델 사용의 장점
  - 이해도 향상
    - » 모델을 사용하려면 실제 개발 범위에 해당하는 복잡한 내용을 단순화해야 하므로 개발 영역을 더 정확히 이해할 수 있음
    - » 프로그램의 전체 모습을 시각적으로 나타낼 수 있으므로 사용자는 관련 요소들의 상호 관계를 빨리 파악할 수 있음
    - » 이해 당사자 간의 의사소통 도구로 활용할 수 있음
  - 유지보수 용이
    - » 추후 요구사항 변경에 따른 유지보수에 활용할 수 있음
    - » 이 후 유사한 시스템을 개발할 때도 활용해 개발 기간과 비용을 줄여줌

- 소프트웨어 개발 모델 사용의 장점
  - 과도한 문서 작업으로 일정 지연
    - » 단계마다 각종 다이어그램을 포함한 문서 작성에 많은 시간을 할애하다 보면 실제 개발 업무가 지연될 수 있음
  - 형식적인 산출물로 전락할 가능성
    - » 단계마다 수많은 문서가 형식적으로만 만들어져 활용되지 못할 수 있음
    - » 변경된 내용을 산출물에 즉시 반영하지 않으면 문서 자체가 사장될 가능성이 높음

04. 요구사항의 표현

■ 모델링

▪ 모델링의 개요

- 앞서 설명한 모델을 제작하는 과정 또는 작업
- 연필로 종이에 간단한 콘티를 작성하거나 스토리보드같이 자기가 표현하고 싶은 것을 문장으로 서술하는 것도 한 종류
- 각 전문 분야의 모델링은 기호나 다이어그램, 표기법 등을 사용해 차이가 없이 일관성 있는 해석이 가능하도록 표현

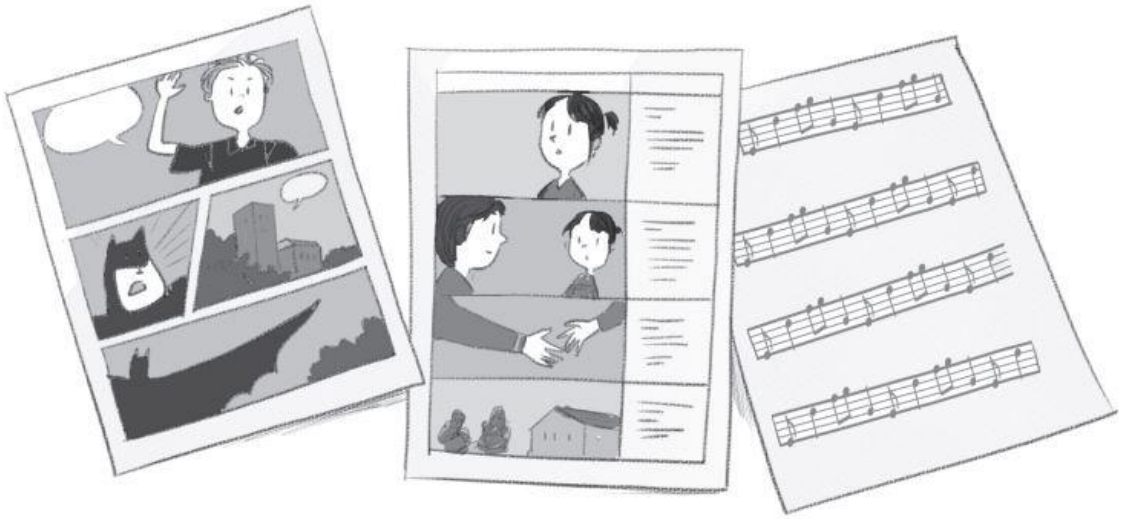

그림 4-12 다양한 형태의 모델링

04. 요구사항의 표현

■ 모델링

- 자연어를 사용한 표현
  - 사용자 요구사항을 자연어로 표현(모델링)하는 것
  - 도구나 기술을 따로 익힐 필요가 없 고 사용자와 대화할 때도 이해하기 쉬움
  - 표현이 길어질 수 있고 해석이 달라질 수 있으며 표현된 내용을 검증하기가 어려움

- 형식 언어를 사용한 표현
  - 친숙하지는 않지만 문법과 의미가 수학을 기초로 작성되어서 간결하고 정확하게 표현할 수 있음
  - 표기법을 별도로 공부해야 하므로 일반 개발자가 사용하기에는 조금 어려운 편
    - » 관계형 표기법: 함의 방정식, 순환 관계, 대수 공리, 정규 표현법을 사용
    - » 상태 위주 표기법: 의사결정 테이블, 이벤트 테이블 , 전이 테이블, 유한 상태 기계, 페트리 넷을 사용

- UML 다이어그램을 사용한 표현
  - UML 다이어그램을 사용하면 개발할 소프트웨어를 가시적으로 볼 수 있고, 개발할 소프트 웨어에 대한 문서화도 가능
  - 이 문서들은 분석 및 설계 과정에서 유용하며 검증 자료로도 활용

04. 요구사항의 표현

■ 모델링 언어

▪ 모델링 언어의 정의

- 모호한 표현의 문제를 해결하기 위해 사용하는 기호, 표기법, 도구
- 악보 기호, 수학 기호 등과 UML 다이어그램, Z 언어와 같은 형식적 표기법 등

• 소프트웨어 개발에서의 모델링 언어

- 요구사항 정의 및 분석·설계의 결과물을 다양한 다이어그램으로 표현하는 표기법
- 모호한 표현이 없 고 일관되어 모델링을 하는 데 매우 유용
- 개발자 간에 원활하게 의사소통할 수 있고 시스템을 주관적으로 해석하는 일도 줄어듬
- 다양한 다이어그램을 사용하므로 시스템을 여러 시각으로 볼 수 있음
- 사용자 요구사항을 검증하는 의사소통 도구로도 사용

• 개발 방법에 따른 모델링 언어

- 구조적 방법: 자료 흐름도(DFD), 자료 사전(DD), 소단위명세서를 사용해 요구분석 결과를 표현
- 정보공학 방법: E-R 다이어그램(ERD)을 데이터베이스 설계 표현을 사용
- 객체지향 방법: 이 방법은 대부분의 나라에서 공통으로 사용하는 UML 표기법을 사용
  UML 표기법은 10여 개의 다이어그램으로 구성되어 있으며 특히 요구사항은 유스케이스 다이어그램을 사용해 표현

1. 자료흐름도의 특징

■ 자료흐름도의 특징

- 도형을 이용한 그림 중심의 표현
- 하향식 분할의 원리를 적용
- 다차원적
- 자료의 흐름에 중점을 두는 분석용 도구
- 제어의 흐름은 중요시 하지 않음

■ 자료흐름도의 작성 효과

- 사용자의 업무 및 요구사항을 쉽게 문서화할 수 있음
- 사용자와 분석가 사이의 의사소통을 위한 공용어 역할을 함
- 일관성 있고 정확한 사용자의 요구사항을 파악할 수 있는 요구분석용 도구의 역할을 수행

2.1 자료흐름도의 4가지 구성요소

■ 처리

- 입력되는 자료흐름을 출력되는 자료흐름으로 변환하는 것

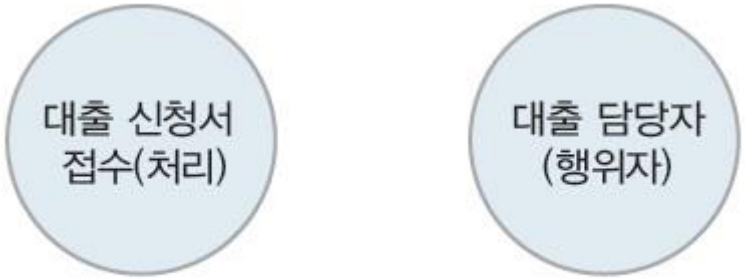

그림 4-1 처리의 표현

■ 자료흐름

- 자료흐름도에서 구성요소들 간의 접속관계를 나타냄

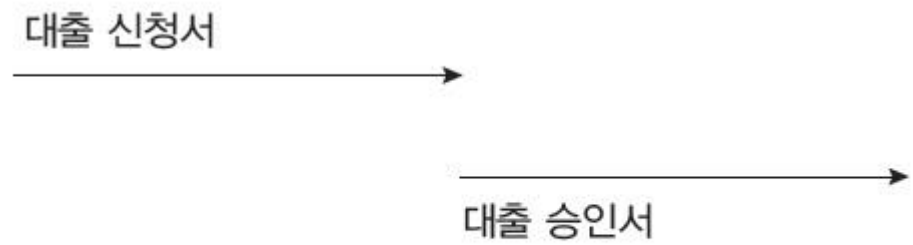

그림 4-2 자료흐름의 표현

2.1 자료흐름도의 4가지 구성요소

■ 자료저장소

- 머물고 있는 자료군의 집합

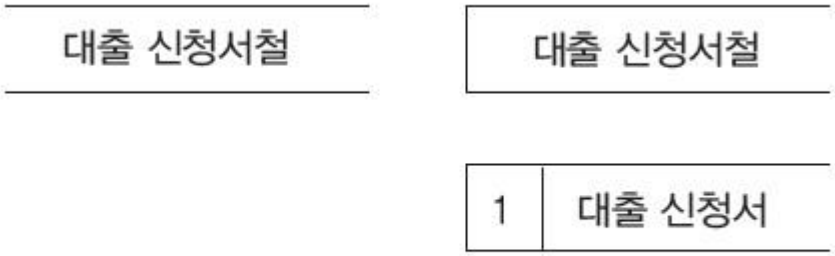

그림 4-3 **자료저장소의 표현**

■ 단말

- 상세한 자료흐름도를 이해할 수 있게 사각형의 단말을 사용함

그림 4-4 **단말의 표현**

2.2 자료흐름도 작성 실습

실습하기 세탁처리 자료흐름도 작성

우선 세탁물을 분류하고, 물에 담가 때를 불린 후 비누질을 하고 헹구어낸 후 (이 과정을 반복할 수도 있다) 탈수해서 말리기까지의 과정이라는 걸 쉽게 분석할 수 있다. 그렇다면 이러한 과정을 새롭게 배운 자료흐름도의 구성요소를 이용해 작성해 보도록 하자.

2.2 자료흐름도 작성 실습

■ 작성 예제 1

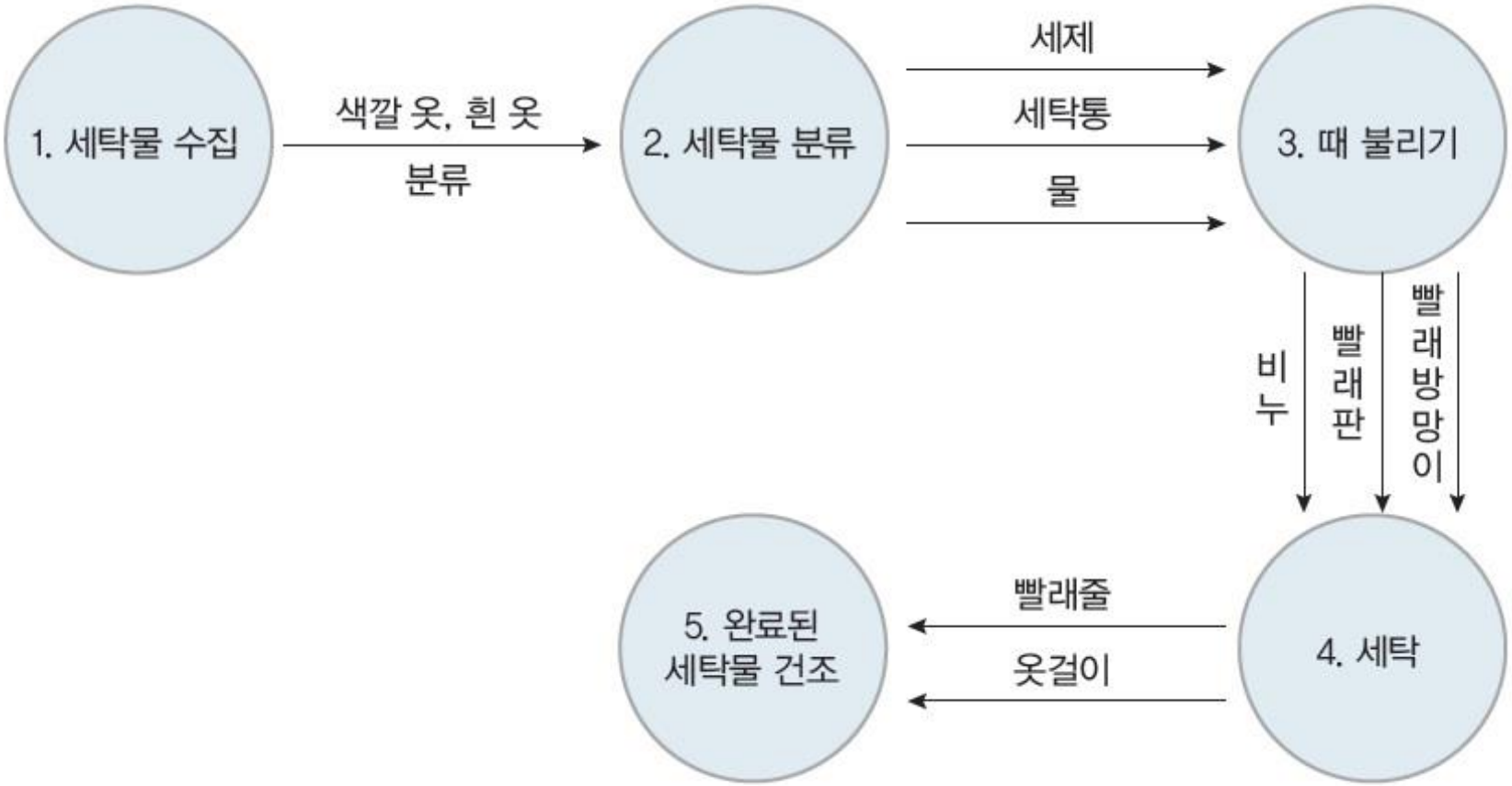

그림 4-5 자료흐름도 작성 예제 1

2.2 자료흐름도 작성 실습

■ 작성 예제 2

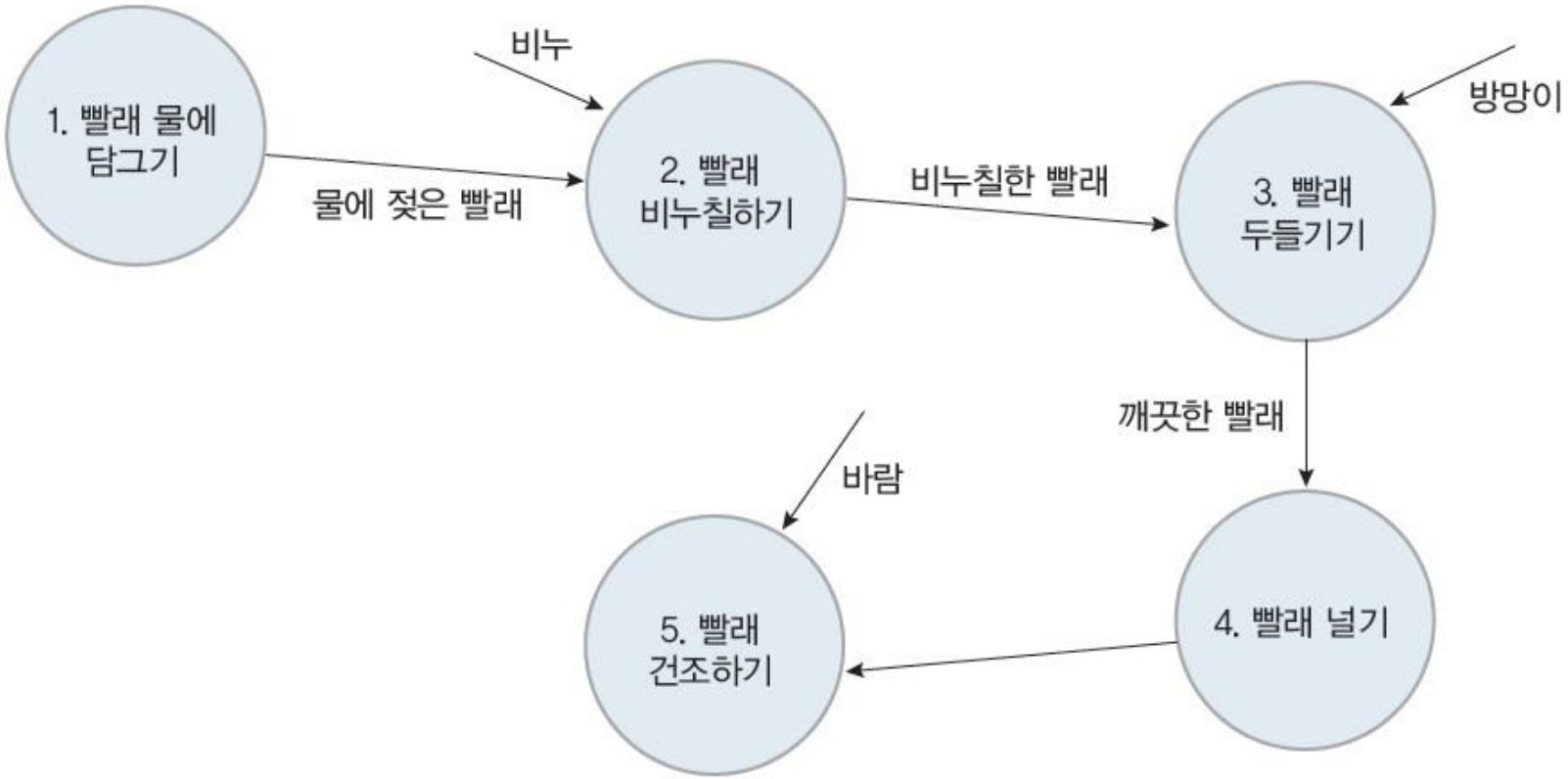

그림 4-6 자료흐름도 작성 예제 2

2.2 자료흐름도 작성 실습

■ 개선된 자료흐름도

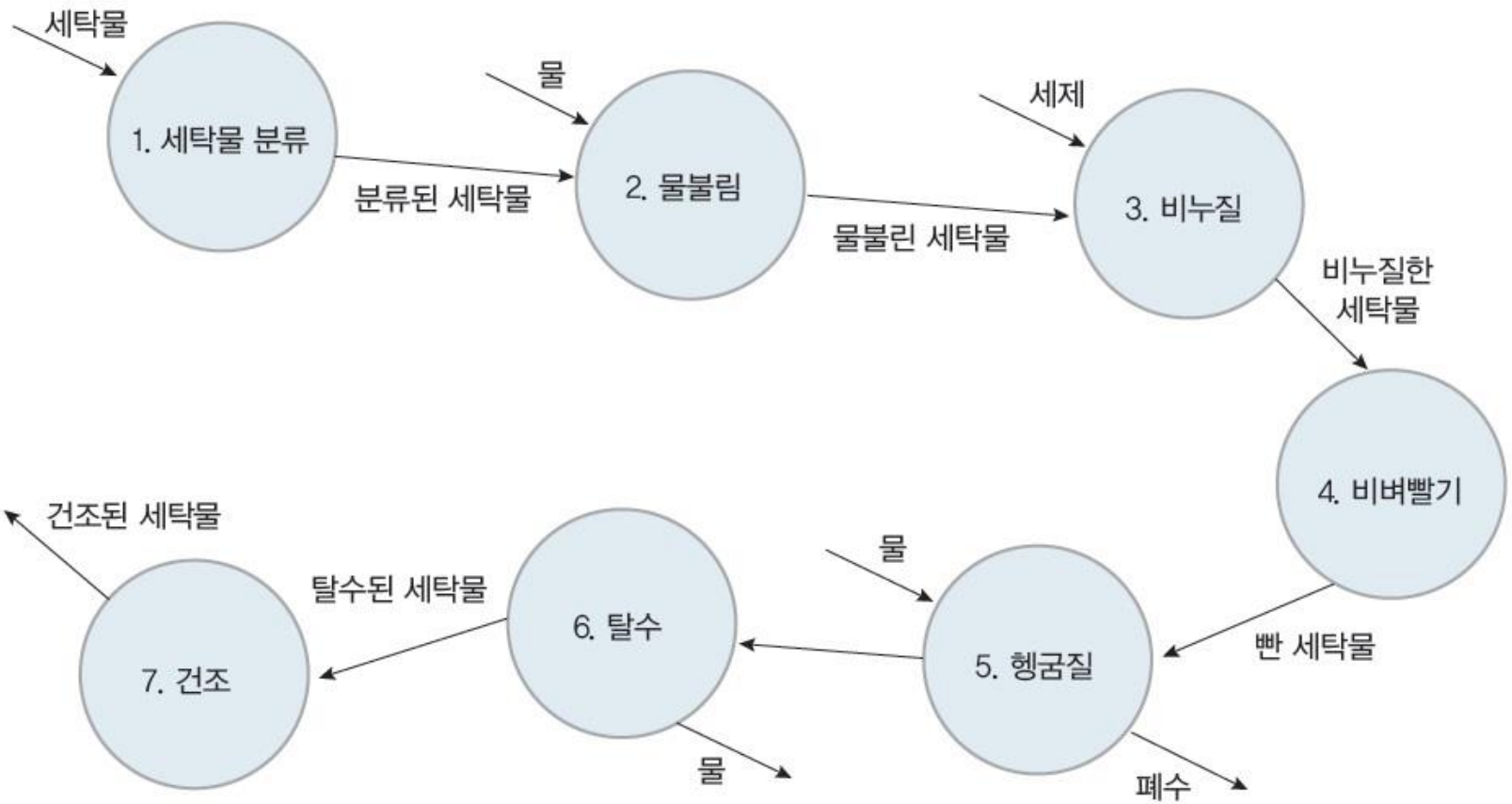

그림 4-7 개선된 자료흐름도

2.2 자료흐름도 작성 실습

실습하기 간호원장 업무 자료흐름도 작성

"저는 이 병원의 간호원장입니다. 제가 업무를 시작하면, 제일 먼저 의사들의 메모철을 살펴 문제점이 있는가를 확인합니다. 문제점이 발견되면 이를 수정하기 위해 의사에게 메모를 보내고, 새로운 치료나 검사할 사항이 있으면 이를 환자철에 기록합니다. 또한 앞으로 할 일을 위해 이 환자철을 보고 검사계획서와 치료계획서를 작성합니다.

제가 환자들을 다 둘러본 후에는, 환자철을 살펴보고 치료가 안 되어 있거나 잘못된 환자가 있는 경우 담당 간호원을 꾸짖고, 의사에게 보낼 메모에 이 사실을 첨부합니다."

2.2 자료흐름도 작성 실습

■ 작성 예제 1

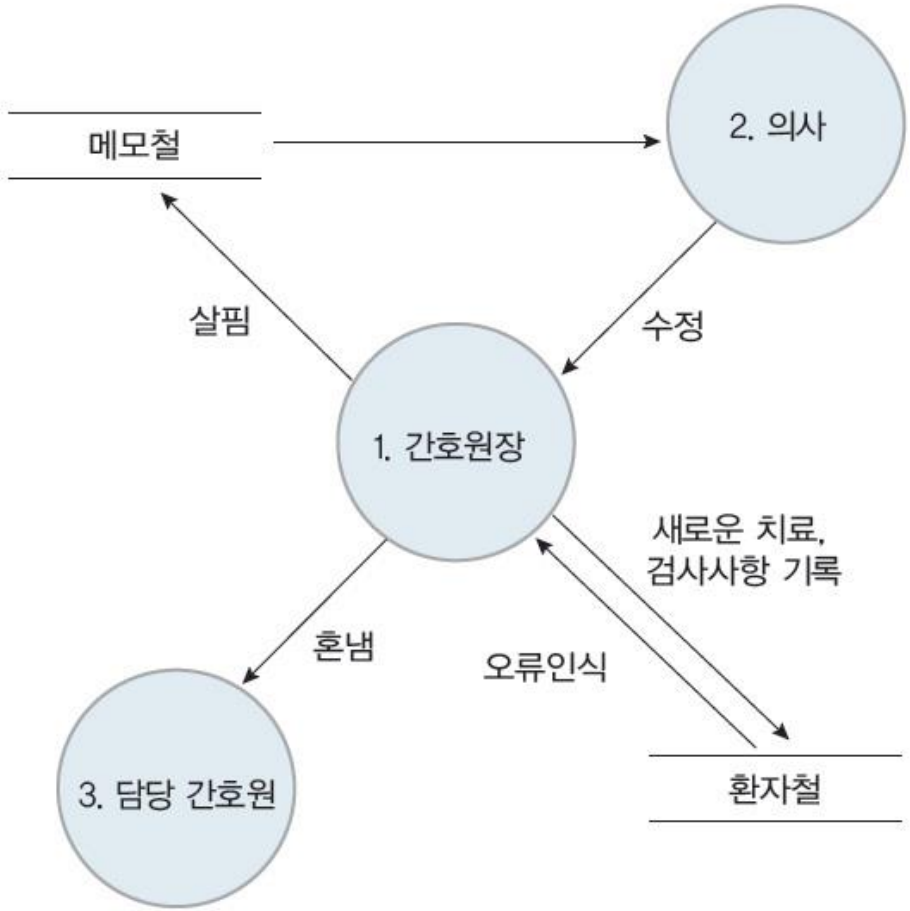

그림 4-8 자료흐름도 작성 예제 1

2.2 자료흐름도 작성 실습

■ 작성 예제 2

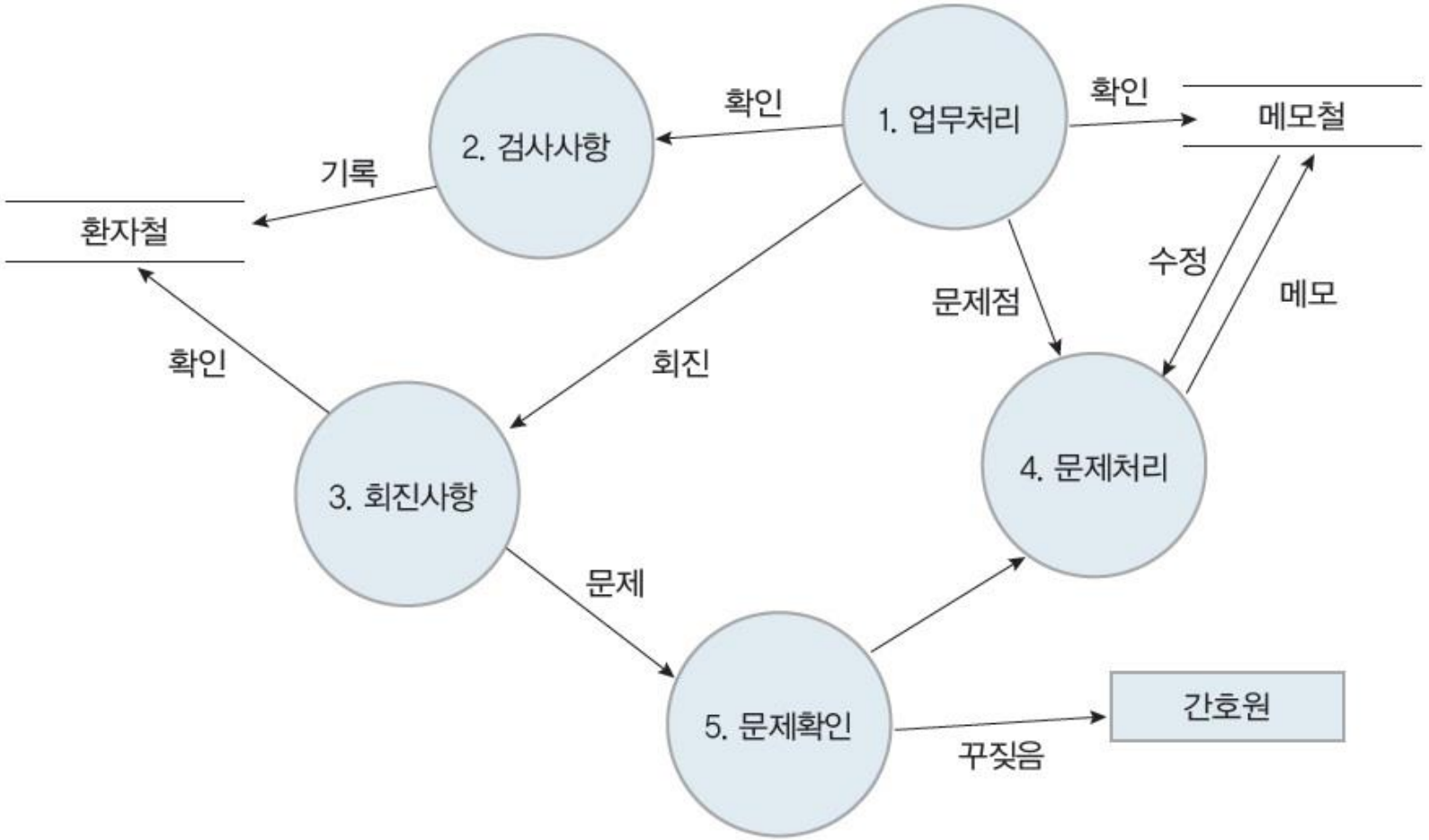

그림 4-9 자료흐름도 작성 예제 2

2.2 자료흐름도 작성 실습

■ 개선된 자료흐름도

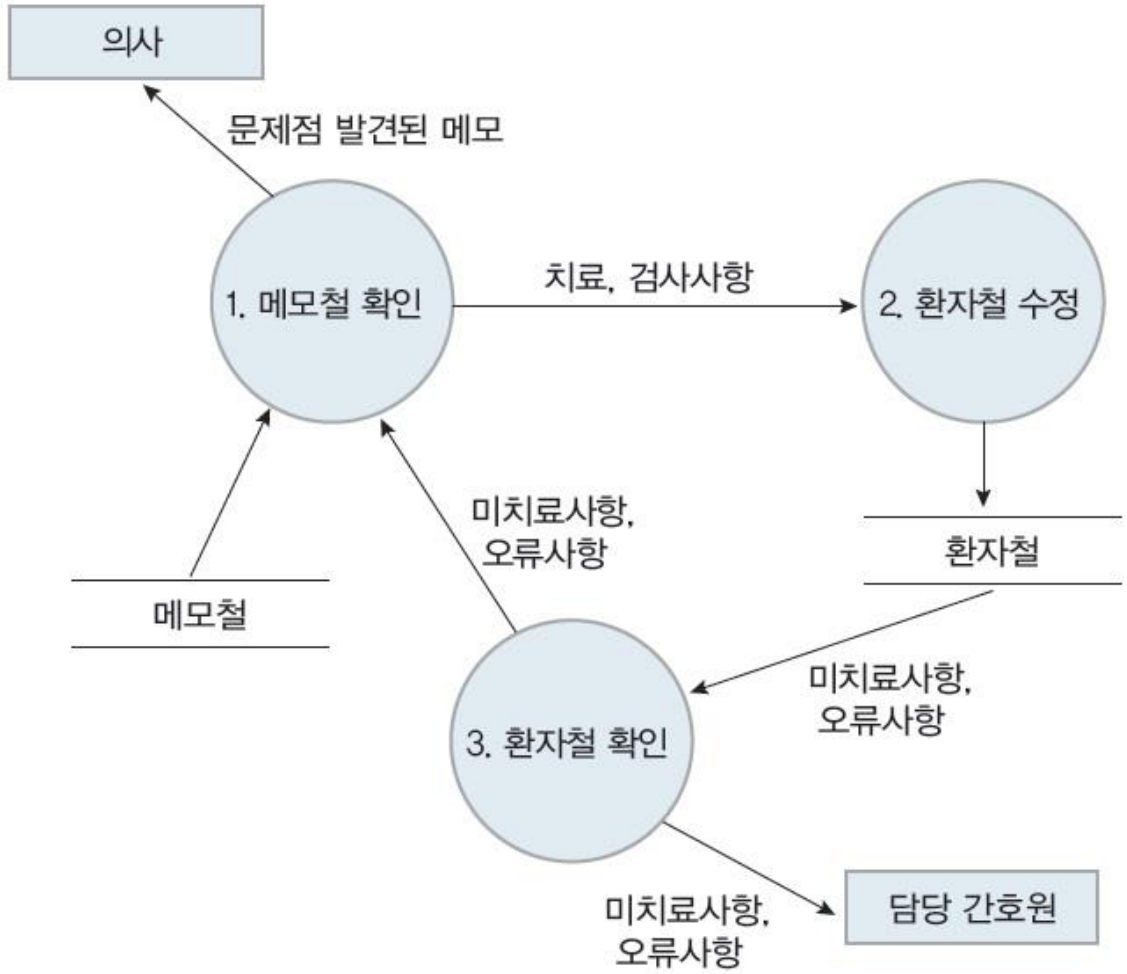

그림 4-10 개선된 자료흐름도

3.1 자료흐름도 작성의 7가지 원칙

① 자료 보존의 원칙(Conservation Rule)

- 어떤 처리의 출력 자료흐름은 반드시 입력 자료흐름을 이용해 생성해야 함

그림 4-11 자료 보존의 원칙에 위배된 예

3.1 자료흐름도 작성의 7가지 원칙

② 최소 자료 입력의 원칙(Parsimony Rule)

- 어떤 처리가 출력 자료흐름을 산출하는 데 반드시 필요로 하는 최소의 자료흐름만 입력해야 하는 것

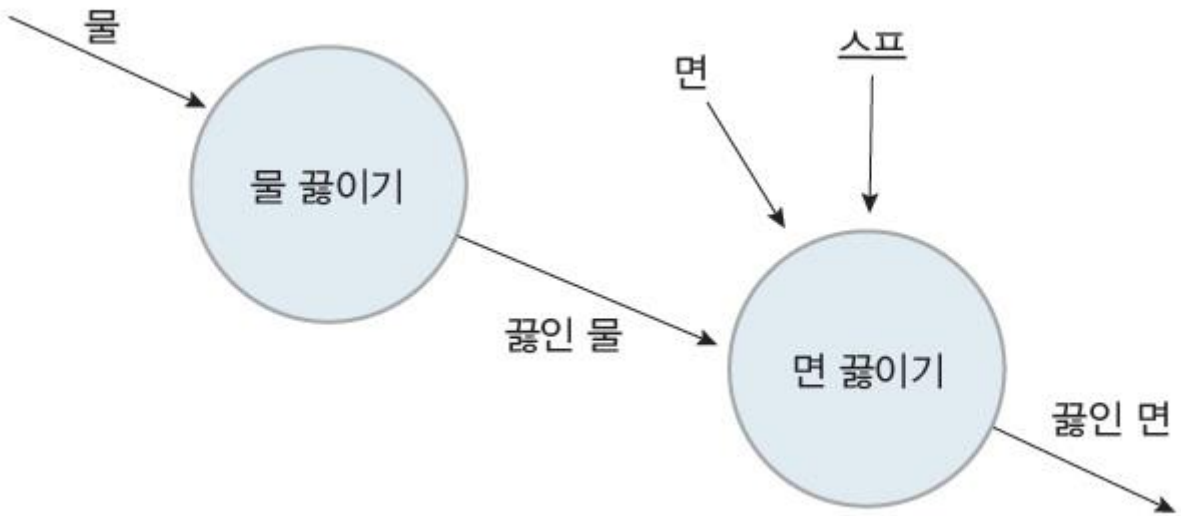

그림 4-12 최소 자료 입력의 원칙 예

3.1 자료흐름도 작성의 7가지 원칙

③ 독립성의 원칙(Independence Rule)

- 자기의 처리는 오직 자신의 입력 자료와 출력 자료 자체에 대해서만 알면 됨
- 독립성의 원칙은 유지보수가 쉬운 시스템을 산출할 수 있음

④ 지속성의 원칙(Persistence Rule)

- 어떤 자료흐름을 기다릴 때를 제외하고는 다시 시작하거나 멈춰서는 안 됨

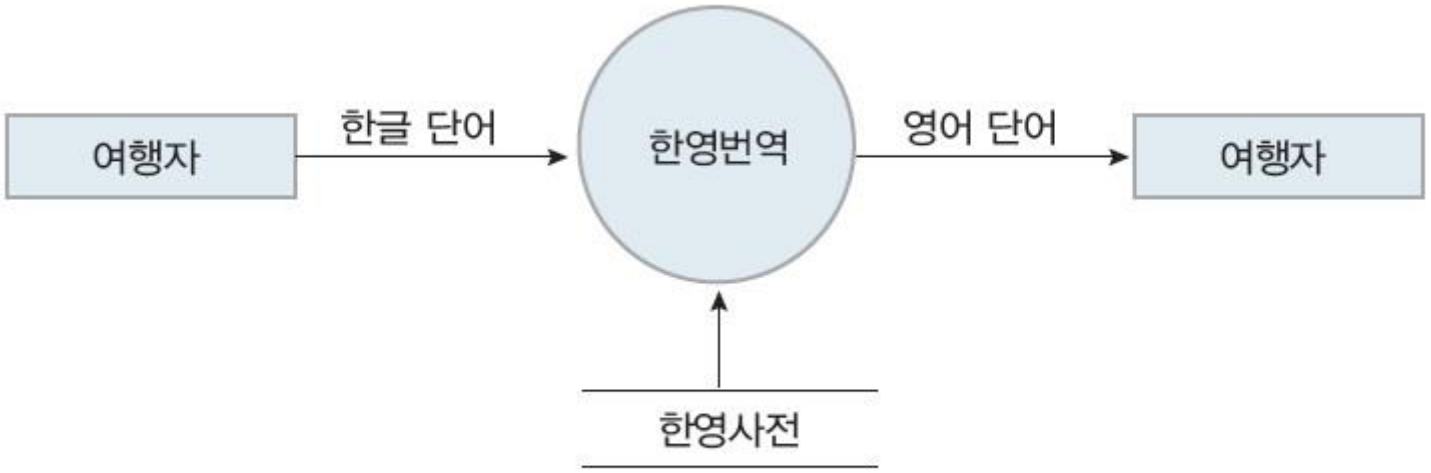

그림 4-13 지속성의 원칙 예

3.1 자료흐름도 작성의 7가지 원칙

⑤ 순차 처리의 원칙(Ordering Rule)

- 처리에 입력되는 자료흐름의 순서는 출력되는 자료흐름에서도 지켜져야 함

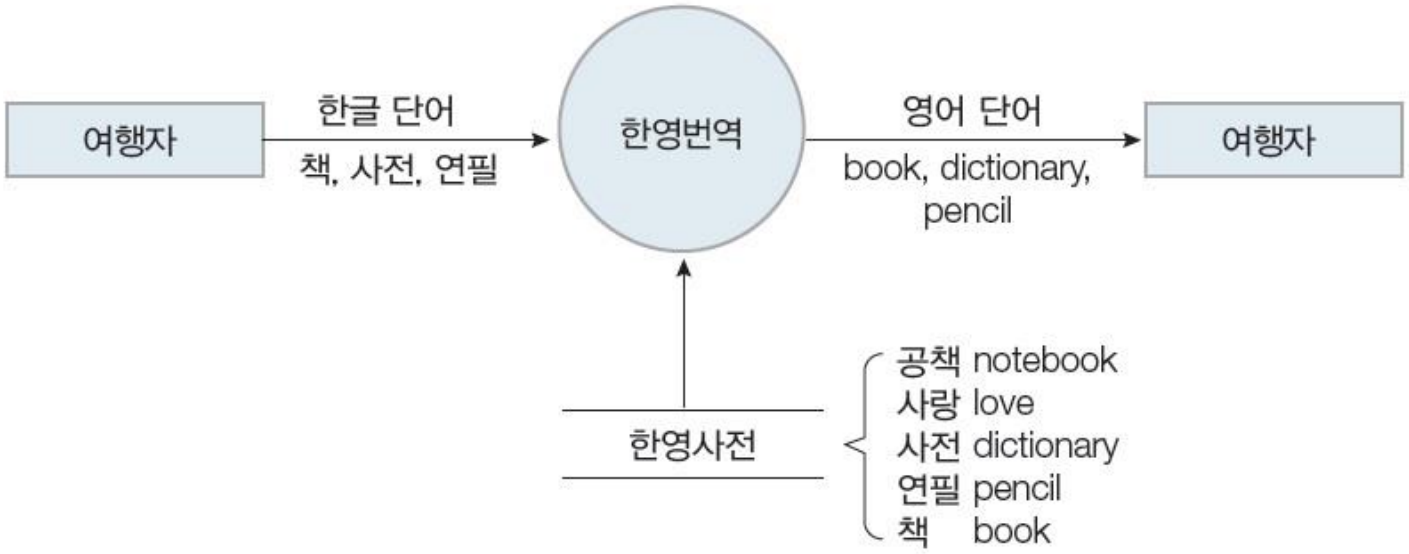

그림 4-14 순차 처리 원칙의 예

⑥ 영구성의 원칙(Permanence Rule)

- 자료저장소의 자료는 제거되지 않아야 함

3.1 자료흐름도 작성의 7가지 원칙

⑦ 자료 변환의 원칙(Nature of Change)

- 자료 본질의 변환
  - 일반적으로 입력 자료흐름에 편집, 계산 등을 해 출력 자료흐름을 산출하는 것

그림 4-15 자료 본질의 변환 예

3.1 자료흐름도 작성의 7가지 원칙

⑦ 자료 변환의 원칙

- 자료 합성의 변환
  - 두 개 이상의 입력 자료흐름에 대해 자료합성의 변환이 발생할 수 있음

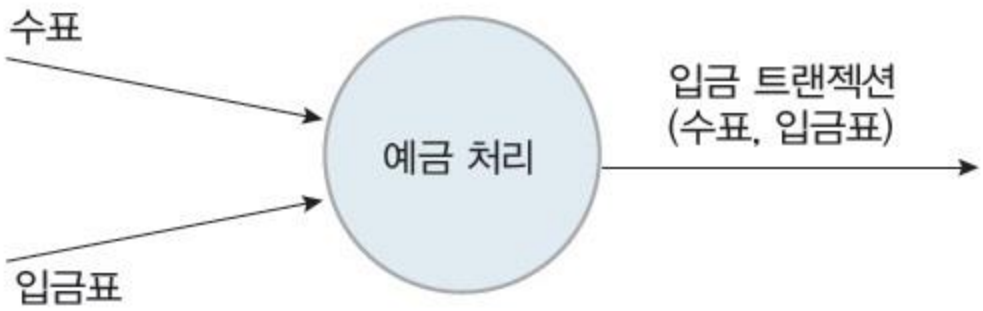

그림 4-16 자료 합성의 변환 예

3.1 자료흐름도 작성의 7가지 원칙

⑦ 자료 변환의 원칙

- 자료 관점의 변환
  - 입력 자료흐름이 동일하게 출력자료흐름으로 나타나게 됨

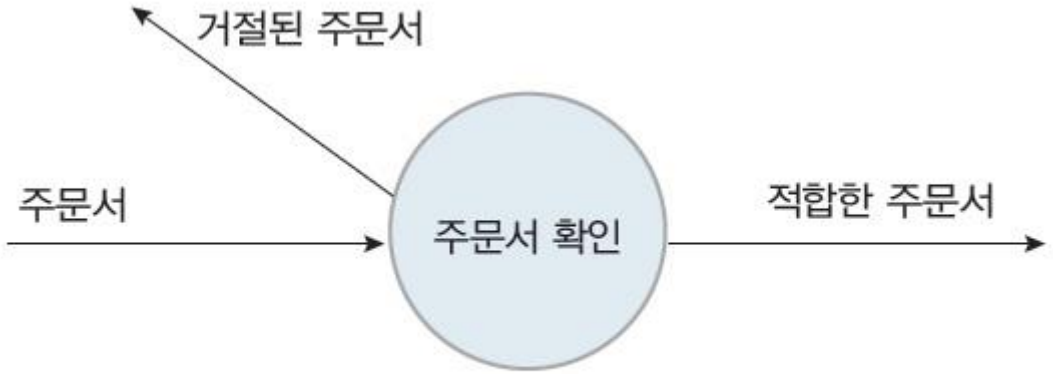

그림 4-17 자료 관점의 변환 예

3.1 자료흐름도 작성의 7가지 원칙

⑦ 자료 변환의 원칙

- 자료 구성의 변환
  - 출력 자료가 입력 자료와 동일하지만, 자료의 구성형태가 변환됨
  - 구성의 변환은 포맷팅 또는 정렬 등을 위한 처리를 필요로 함

그림 4-18 자료 구성의 변환 예

4.1 자료흐름도의 작성 절차

① 시스템 경계의 입출력 식별

- 순수 입력과 출력을 선정하는 것은 분석의 대상이 무엇이어야 하는가에 대한 결정과 관련됨

② 시스템 경계 내부의 작성

- 자료흐름도를 작성하는 것은 현재 사용자 영역을 최초로 문서화하는 것
- 현재의 업무수행 방식을 그대로 기술해야 함

③ 자료흐름의 명명

- 각각의 자료흐름에 대해 새로운 명칭을 부여
- 명칭을 부여할 때는 전체의 자료흐름에 적용될 수 있는 이름을 부여
- '자료', '정보' 등과 같이 의미 없는 명칭은 부여하지 않음
- 전체로 통합될 수 없는 자료항목을 하나의 자료흐름으로 만들지 않음

4.1 자료흐름도의 작성 절차

③ 자료흐름의 명명

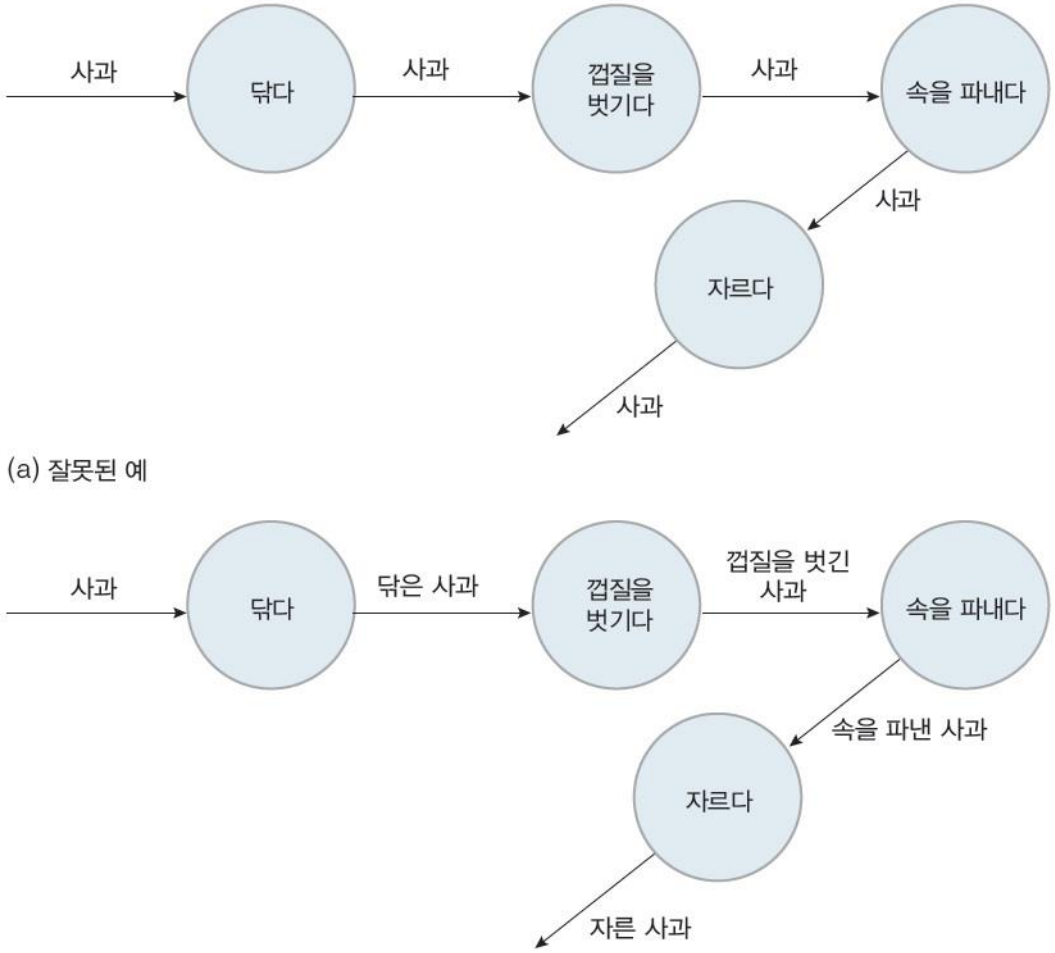

그림 4-19 자료흐름의 명명

4.1 자료흐름도의 작성 절차

④ 처리의 명명

- 하향식 접근방법 : 자료흐름에 중점을 두어 명명하고, 처리에 관한 명명하는 것
- 상향식 접근방법 : 처리에 먼저 중점을 두는 접근방식

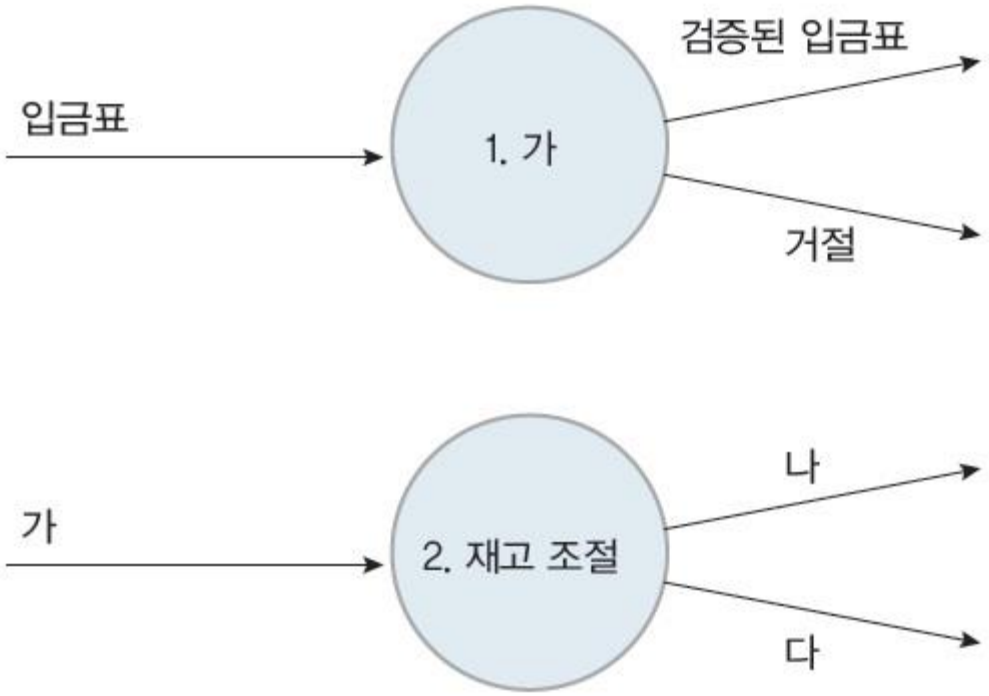

그림 4–20 처리의 명명

4.1 자료흐름도의 작성 절차

④ 처리의 명명

- 처리의 명칭은 처리내용에 적합하도록 명명해야 함
- 처리의 이름은 동사형 명사와 단일 직접목적어를 사용함
  (두 개의 동사가 필요하다면 처리를 분할해야 함)
- 어떤 경우에도 다 적용될 수 있는 포괄적인 명칭은 피해야 함
- 명칭부여가 불가능한 처리가 없도록 분할함

4.2 자료흐름도 작성 시 주의사항

- 초기화와 종료화는 고려대상에서 제외
- 사소한 오류처리는 생략
- 제어흐름은 표시하지 않음

4.3 자료흐름도 검토 및 개선

자료흐름도 검토 및 개선

- 주어진 일을 불완전하게 실행한 후, 그것에 대해 개선하는 방식이 좋음
- 대체할 때마다 개선이 이루어지고 더 나은 자료흐름도가 완성됨

Thank You