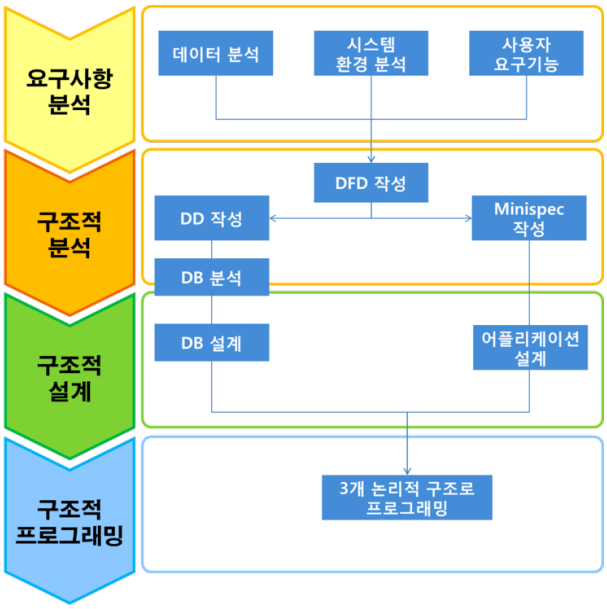
소프트웨어 개발 방법론

구조적 방법론

- 절차 지향 소프트웨어 개발 방법론
- 제한된 구조에서 코드 생성 및 순차적 실행
- 프로세스 단위로 문제 해결 -> 복잡하고 큰 시스템을 작고 독립적인 서브 시스템으로 나눠 시스템을 쉽게 해결할 수 있음



1) 요구사항 분석 : 고객이 원하는 요구사항을 끌어내 명세화하는 것

2) 구조적 분석 : 고객이 원하는 기능/시스템환경/데이터를 종합하여 데이터 흐름도(Data Flow Diagram)를 작성

3) 구조적 설계 : 모듈 중심 설계 단계-> 목적 : 재활용, 결합도를 낮춰 독립성을 높임

4) 구조적 프로그래밍 : 순차, 선택, 반복의 논리 구조 구성으로 프로그램 복잡성 최소화

1. 요구사항 및 구조적 분석

- 구조적 분석의 필요성

· 시스템을 분할하여 분석하고 도형 중심의 문서화 도구를 사용함으로써 분석자와 사용자 간의 의사소통이 용이하다

· 전체 시스템을 일관성 있게 이해할 수 있고, 시스템을 하향식으로 세분화하므로, 분석의 중복성을 배제할 수 있다.

- **구조적 분석 도구**

1) DD(Data Dictionary)

자료의 의미나 자료의 단위 및 값에 대한 사항을 정의하여 자료 속성을 파악할 수 있다.

ex) Dialog data

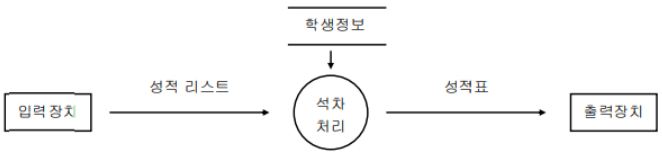
대화 데이터는 크게 A음식점, B의류, C학원, D소매점, E생활서비스, F카페, G숙박업, H관광여가오락으로 나누어져 있음.

Dialogue data = Speaker [고객 | 점원] + Sentence + dataID + domainID + Category + SpeakerID[0 | 1] + SentenceID + Main + (Sub) + QA [Q | A] + (QACNCT) + MQ + SQ + UA + SA + (개체명) + (용어사전) + (지식베이스)

2) DFD (Data Flow Diagram)

각 기능을 최대한 상세화가 될 때까지 분할하여 표현한 구조

데이터에 비해 기능이 복잡하고 중요할 경우에 유용하게 사용됨.

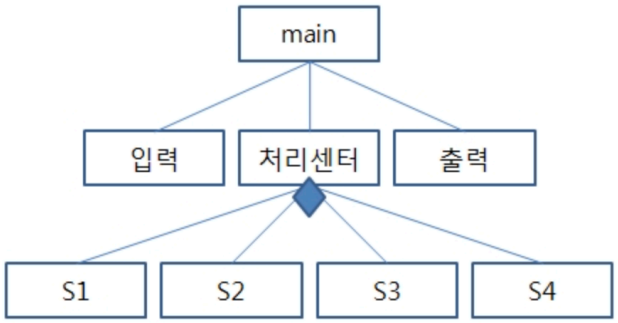


□	자료의 출처 및 도착지(Terminator)
○	프로세스, 기능, 변환, 버블
→	자료의 흐름
=	자료 저장소

시스템을 모듈 단위로 분할하여 모듈의 계층적 구성과 입출력 인터페이스를 도출함.

크게 입력 → 변환센터 → 출력 으로 나눈다.

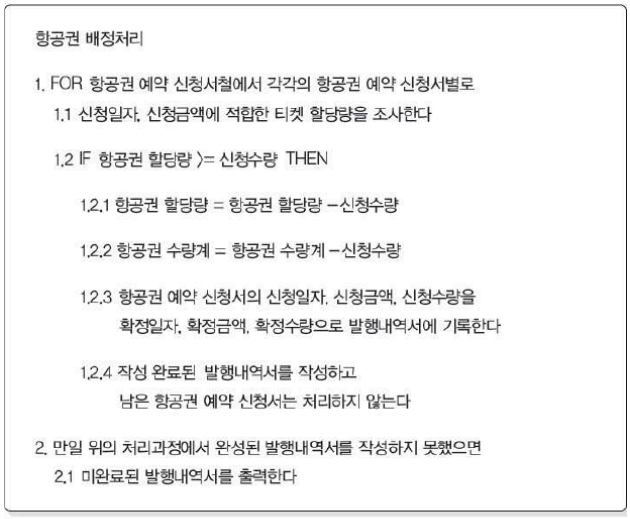
변환 센터에서 여러 개의 자료 흐름이 일어남.



3) 소단위 명세서(Mini-Spec)

입력 자료를 출력 자료로 변환하기 위해 수행되어야하는 정책이나 규칙을 구체적으로 기술하는 도구

소단위 명세서는 구조적 언어, 의사 결정 테이블(판단표), 의사 결정도를 이용하여 기술함.

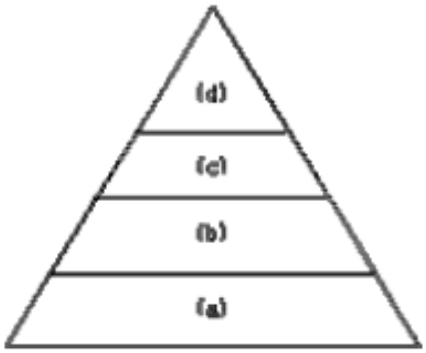


항공권 배정처리

1. FOR 항공권 예약 신청서철에서 각각의 항공권 예약 신청서별로
   1.1 신청일자, 신청금액에 적합한 티켓 할당량을 조사한다
   1.2 IF 항공권 할당량 >= 신청수량 THEN
      1.2.1 항공권 할당량 = 항공권 할당량 – 신청수량
      1.2.2 항공권 수량계 = 항공권 수량계 – 신청수량
      1.2.3 항공권 예약 신청서의 신청일자, 신청금액, 신청수량을
         확정일자, 확정금액, 확정수량으로 발행내역서에 기록한다
      1.2.4 작성 완료된 발행내역서를 작성하고
         남은 항공권 예약 신청서는 처리하지 않는다
2. 만일 위의 처리과정에서 완성된 발행내역서를 작성하지 못했으면
   2.1 미완료된 발행내역서를 출력한다

그림 3-4 전형적인 소단위 명세서

2. 구조적 설계



- (d) 절차설계
  - 모듈이 수행할 기능을 절차적 기술로 바꾸는 것
- (c) 인터페이스 설계
  - 시스템과 사용자가 어떻게 통신하는지
- (b) 구조 설계
  - 모듈간의 관계와 프로그램 구조 정의
- (a) 데이터 설계
  - 요구사항 분석 단계에서 생성된 정보를 S/W를 구현하는데 필요한 자료구조로 변환하는 것

3. 구현

설계 단계에서 생성된 내용을 컴퓨터가 알 수 있는 형태로 변환하는 과정

구조적 프로그래밍 (순차, 선택, 반복)

4. 검사(테스트)

품질 보증 활동의 하나로 오류를 발견하기 위해 수행하는 과정

* 화이트 박스 테스트

모듈 안의 작동을 자세히 관찰, 제품의 내부 요소가 명세서에 따라 수행되고 실행되는지 보장

원시 코드의 모든 문장을 한번 이상 수행, 각 조건에서 참과 거짓 모든 논리적 결정이 적어도 한번 이상 실행함

* 블랙박스 테스트

s/w가 수행할 특정 기능을 알기 위해 각 기능이 완전히 작동되는 것을 입증하는 방법

해당 입력 자료에 맞는 결과가 출력되는지, 입력 조건의 경계값에서 오류가 날 확률을 확인