

소프트웨어 공학개론

Introduction to Software Engineering

2022-1학기 (Spring)

선문대학교 AI소프트웨어학과

3월 29일 (화)

Part I. 개념

- 4. 요구사항분석
 - 4.1 요구사항분석이란?
 - 4.2 요구사항획득
 - 4.3 요구사항의 사양화
 - 4.4 요구사항 확인
- 연습 문제

개요

- 소프트웨어개발의 목적
 - 고객과 이용자가 만족하는 소프트웨어를 구현
 - 고객과 이용자의 요구사항을 정확하게 파악하는 것이 중요
- 요구사항분석의 개요에 대해서 설명
 - 요구사항획득
 - 요구사항 사양화
 - 요구사항확인

개요

- **요구사항분석(요구사항정의)**
 - 고객 및 이용자의 요구사항을 찾아내서 사양화하는 작업
 - 개발해야하는 소프트웨어시스템에 대해서 고객과 이용자가 명시적 또는 잠재적으로 생각하고 있는 요구사항을 도출하여 시스템 전체의 사양을 가능한한 엄밀하게 정의하는 작업
 - 소프트웨어개발공정의 최상위에 위치
 - 요구사항분석 공정의 성과물은 이후의 각 공정들에 많은 영향을 줌
- **요구공학(Requirement Engineering)**
 - 소프트웨어의 요구사항에 관한 기술을 집대성한 학문분야

목차

- 요구사항분석의 개요
 - 4.1.1 요구사항과 요구사항
 - 4.1.2 요구사항분석 작업의 관계자
 - 4.1.3 요구사항분석의 어려움
 - 4.1.4 요구사항분석 작업

4.1.1 요구사항과 요구사양

- 요구사양
 - 요구사항을 만족시키기 위해 소프트웨어가 구현해야하는 요건들을 모아서 정리한 것
 - **요구사항**
 - 소프트웨어를 이용함으로써 실현하려는 내용을 모아서 정리한 것
 - 소프트웨어를 이용하는 사용자쪽에서 보면 소프트웨어가 구현해야되는 목표
 - 요구사항 분류
 - **기능적 요구사항** : 개발하려는 시스템이 무엇을 수행하는지를 표현
 - **비기능적 요구사항** : 성능, 사용편의성, 안전성, 유지보수성, 이식성 등을 표현
 - 서로 충돌하는 경우가 많음
 - 요구사항에 우선순위를 정해서 장단점을 비교분석하는 등의 대책 마련 필요
 - **사양**
 - 요구사항을 구현하기 위해 필요한 기능과 성능을 제공할 때의 조건 및 제약사항
 - 요구사양서
 - 요구사항분석에 의해 요구사양을 문서화한 것

4.1.1 요구사항과 요구사양

- 이용자와 고객이 요망하는 요구사항이 반영되어 있지 않거나 요구사항의 일부분이 누락되면
 - 설계공정 및 구현공정에서 전혀 다른 소프트웨어가 작성되는 사태 발생
 - 개발종료 후에 소프트웨어에 대한 쓸모없는(불필요한) 수정작업 재실시
 - 설계작업 및 구현작업 재실시
 - 개발비용의 대폭적인 증가와 개발일정의 지연
- 개발 초기단계에서 본질적인 요구사항을 정확하게 정의가능하다면
 - 이용자와 고객이 원하는 시스템을 구축할 수 있는 가능성이 높아짐
 - 개발에 있어 재작업 회피가능

4.1.2 요구사항분석 작업의 관계자

- (1) 고객 및 이용자
- (2) 개발자
- (3) 분석가(System Analyst)

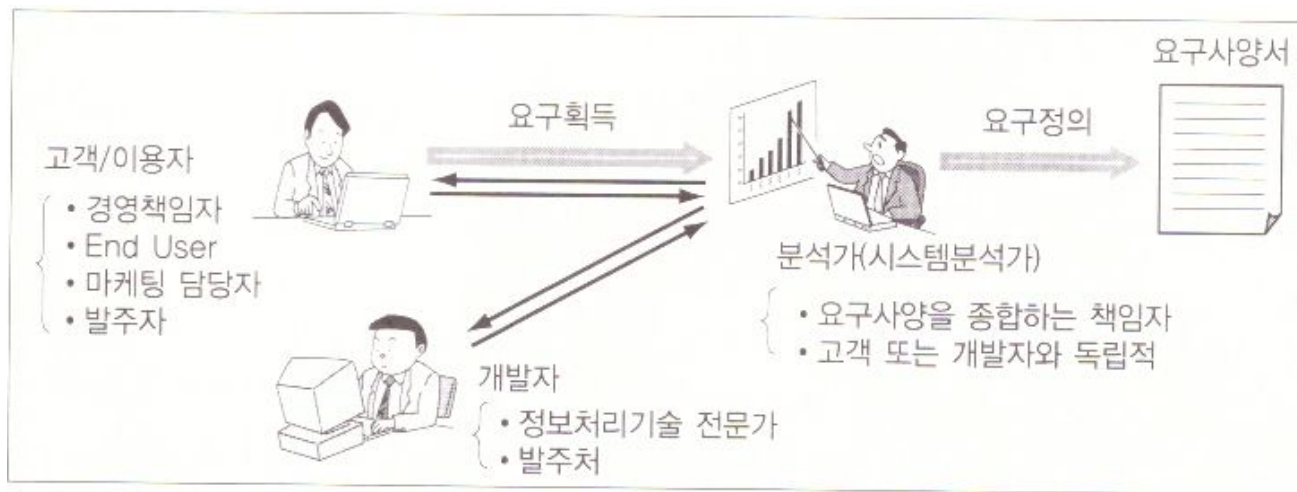


그림 4.1 요구사항분석작업의 관련자

4.1.2 요구사항분석 작업의 관계자

- (1) 고객 및 이용자
 - 고객
 - 개발하려는 시스템을 필요로 하는 조직
 - 기업에서 요구사항의 결정과 개발시스템의 인수작업에 관한 권한을 갖는 사람
 - 발주자: 시스템의 전체 또는 일부분을 외주에 의해 개발하는 경우
 - 마케팅 담당자: 일반 소비자를 대상으로 하는 소프트웨어제품을 개발하는 경우
 - 이용자(End User)
 - 실제로 개발하려는 시스템을 사용하여 업무를 수행하는 사람
 - 조직 및 기업에서 시스템을 관리하는 운용자

4.1.2 요구사항분석 작업의 관계자

- (1) 고객 및 이용자
 - 고객과 이용자가 같은 사람인 경우와 다른 사람인 경우가 있음
 - 고객과 이용자가 서로 다른 경우 주의 필요
 - 고객(특히 기업의 책임자): 개발하려는 시스템이 발휘하게 되는 이익과 개발비용을 강하게 의식하는 경우가 많음
 - 이용자: 소프트웨어에 대한 사용편의성을 요구하고 개발비용에 대해서는 신경쓰지 않음
 - 관점(Viewpoint)을 의식하여 고객과 이용자의 요구사항에 관해 상호합의 필요
 - 관점(Viewpoint): 요구사항을 찾아낼 때의 입장이나 견해

4.1.2 요구사항분석 작업의 관계자

- (2) 개발자
 - 정보처리기술 전문가로서 실제로 시스템을 개발하는 사람
 - 기술적인 관점에서 신기술에 의해 구현가능한 요구사항을 제안 또는 비현실적인 요구사항 배제
 - 정보시스템 담당부서에 소속된 인력: 자체개발 하는 경우
 - 발주처: 외주에 의한 개발인 경우

- (3) 분석가(System Analyst)
 - 실제로 요구사항분석을 수행하여 요구사항을 모아서 정리하는 책임자
 - 요구사항분석에서 가장 중요한 역할 수행
 - 이용자와 개발자 이해관계를 조정(의사소통에 있어 간격을 메움)
 - 개발시스템이 취급하는 업무와 시스템개발에 대해서도 정통해 있어야 함
 - 이용자와 개발자 양쪽에 독립적인 입장 유지하는 것이 바람직

4.1.3 요구사항분석의 어려움

- (1) 고객과 이용자의 요구사항이 애매모호
 - 고객과 이용자가 다른 경우
 - 고객이 이용자의 업무를 이해 못함
 - 이용자가 업무내용 및 업무에 관한 문제점을 정확하게 파악 못함
 - 개발하려는 시스템에서 첨단기술 이용하는 경우
 - 현행 업무내용이 대폭 변경 가능성 높음
 - 진정한 이용자가 누구인지를 결정하는 것이 결정하는 것이 어려움
 - 누구로부터 요구사항을 도출해 내면 좋은지 알 수 없음

4.1.3 요구사항분석의 어려움

- (2) 고객과 이용자의 요구사항 변화
 - 새로운 요구사항이 있는 경우
 - 요구사항분석 수행 도중, 시장과 사회정세, 고객의 환경이 변화하는 경우
- (3) 이용자와 개발자 사이의 원활하지 못한 의사소통
 - 서로 다른 지식
 - 개발자
 - 시스템개발에 관한 지식 풍부
 - 그러나, 개발시스템이 이용되는 업무에 관한 지식(도메인) 결여
 - 이용자
 - 업무에 관한 지식 풍부
 - 그러나 시스템개발에 관한 지식 부족
 - 각각 사용하는 전문용어가 다름
 - 같은 요구사항을 보더라도 각각의 해석이 크게 다르게 되거나 한쪽의 해석이 잘못될 가능성이 있음

4.1.4 요구사항분석 작업

- (1) **요구사항획득**
 - 고객 및 이용자가 진정으로 요구하는 사항을 도출하여 요구사항으로 종합정리하는 작업
 - 관계자들의 목소리에 귀를 기울이거나 업무를 관찰함으로써 수행
 - 요구사항기술
 - 요구사항획득에 의해 얻어진 요구사항을 정리한 것
- (2) **요구사항의 사양화**
 - 요구사항을 완성시키는 작업
 - 요구사항기술로부터 오류 및 불필요한 부분을 제거
 - 애매모호한 부분을 배제
 - 부족한 정보를 보충
 - 자연언어, 도면, 형식언어 등 사용

4.1.4 요구사항분석 작업

- (3) 요구사항확인
 - 작성된 요구사항이 올바른지 확인하는 작업
 - 수학적 증명에 의해 요구사항 검사가능
 - 요구사항이 형식적으로 기술되어 있는 경우
 - 요구사항 리뷰
 - 사양의 일관성, 완전성, 정확성 등에 관한 체크리스트를 사용하여 요구사항서를 담당자가 검사하는 작업

개요

- 시스템개발이 수행될 때에는 해당시스템에 의해 해결되어야하는 문제 존재
 - 문제가 복잡하거나 애매모호한 경우 문제의 정확한 인식 어려움
- 다른 개발과정에서의 작업과 다르며, 상류과정으로부터 주어지는 정보가 없음
- 일반적인 문제분석기법이 적용되는 경우가 많음

목차

- 4.2.1 요구사항의 추출
- 4.2.2 요구사항의 취사선택

4.2.1 요구사항의 추출

- 자주 사용되는 **요구사항 추출기법**
 - 목적이 요구사항 추출만은 아니며 추출된 요구사항에 관한 분석 및 모델화를 포함
 - 구체적인 절차가 불명확한 것도 많음
 - (1) 자료수집
 - 시스템과 해당 업무에 관한 기존 자료들을 수집, 분석함으로써 요구사항 추출
 - 현행시스템이 존재하는 경우
 - 해당 시스템의 동작을 이해함으로써 문제점을 명확히 함
 - (2) 인터뷰
 - 특정한 고객 및 이용자를 선택하여 시스템에 관한 질의를 통해서 요구사항 청취
 - 어느 정도의 결정권을 가지고 있고 업무에 정통해 있는 인물 선택여부가 중요
 - (3) 설문조사
 - 고객과 이용자에 대한 질문사항을 미리 가정하는 경우 유효함
 - 준비하는 질문사항에 따라 조사결과가 한정적으로 될 가능성이 있음에 주의

4.2.1 요구사항의 추출

- 자주 사용되는 요구사항 추출기법
 - (4) **브레인스토밍 (Brainstorming)** - 참고: 엘리트주의, 집단지성
 - 관계자들을 업무내용 등을 기준으로 여러 그룹들로 나누어서 시스템에 관한 자유로운 의견을 이야기 나눌 수 있는 기회를 제공하고 이러한 대화로부터 요구사항 추출
 - 보다 많은 자유로운 아이디어와 타인의 의견을 발전시킨 아이디어가 환영받음
 - 타인의 발언을 부정해서는 안됨
 - 수집된 의견의 분석 및 정리에는 **KJ기법**을 이용가능
 - **KJ기법**
 - 일본 동경공업대학 **Kawakita Jiro**교수 창안
 - 개개의 사실이나 정보를 상관관계가 있는 것끼리 그룹핑하여 도식화하는 방법
 - 각각의 데이터를 별개의 카드에 기입하고, 각 카드에 대한 그룹화를 반복함으로써 그룹들간의 관계 정리
 - 회의중의 발언이나 시장조사에서의 정보 등을 따로따로 카드에 기입한 후에 연관성이 있는 것을 정리/배열함으로써 문제를 발견하고 아이디어 발상을 수행하는 기법

4.2.1 요구사항의 추출

- 자주 사용되는 요구사항 추출기법
 - (5) 현장관찰
 - 실제로 업무를 체험하거나 현장업무를 직접 관찰함으로써 현장에서 발생하는 문제점 조사
 - 현장작업자에게 큰 부담을 주지 않도록 사전에 업무에 관한 내용에 대해서 충분히 이해해 둘 필요가 있음
 - (6) 프로토타이핑
 - 요구사항을 토대로 시스템의 프로토타입을 작성하여 고객과 이용자에게 제시함으로써 기능과 사용편의성을 부분적으로 평가받음
 - 프로토타입 작성은 소프트웨어 개발기술이 필요하므로 프로토타입 작성 자체가 어려운 문제꺼리가 되는 경우가 있음

4.2.1 요구사항의 추출

- 자주 사용되는 요구사항 추출기법
 - (7) 시나리오
 - 특정한 상황에서 이용자가 목표를 달성하기 위해 수행하는 행동과 그 행동에 의해 얻어지는 이미지를 시간흐름에 따라서 기술한 것
 - 자연언어, 도면, 애니메이션, 동영상 등으로 표현
 - 이용자와 개발자가 공동으로 시나리오를 작성
 - 개발시스템을 누가 어떻게 이용하는지에 대한 이미지가 초기단계에서 명확하게 됨
 - 보다 구체적인 요구사항을 이용자로부터 도출하기 수월해짐
 - 시나리오가 매우 구체적이기 때문에 발생하는 문제점
 - 시나리오 내용이 대상자에게 크게 의존함

4.2.1 요구사항의 추출

- 자주 사용되는 요구사항 추출기법
 - (8) 유즈케이스 (Use Case)
 - 이용자가 어떻게 시스템을 사용하는지를 나타내는 전형적인 사례
 - 시스템과 시스템외부 사이의 경계를 명확하게 분리
 - 시스템의 각 기능마다 시스템과 외부와의 상호작용을 기술
 - 시나리오를 일반화한 것
 - 시나리오 : 이용자와 시스템 사이의 대화를 직접 표현

4.2.1 요구사항의 추출

- 자주 사용되는 요구사항 추출기법
 - (9) 목표지향분석
 - 시스템을 개발하는 것은 어떤 목표를 달성하기 위해서라는 생각을 토대로 하는 기법
 - 맨 처음에는 달성해야하는 커다란 목표를 명확하게 하고, 그 목표를 부분 목표들로 분해하는 과정을 반복함으로써 **Goal Tree** 작성
 - 목표 2종류에 의해 계층화
 - AND분해 : 분해되는 부분목표가 모두 만족되면 상위의 목표가 만족
 - OR분해 : 부분목표들 중 한 개가 만족되면 상위의 목표가 만족
 - 충분히 상세화된(계층구조의 최하위에 위치) 부분목표들이 요구사항의 후보가 됨
 - 장점: 목표와 요구사항이 체계적으로 정리됨
 - 단점: 목표를 달성하는 것에 특화된 요구사항만 도출

4.2.1 요구사항의 추출

- 자주 사용되는 요구사항 추출기법
 - (9) 목표지향분석
 - 목표분할만을 반복함으로써 작성된 Goal Tree에서 상위목표와 하위목표가 상호모순되는 경우가 있음
 - 그림 예) “인증없음”이라는 하위목표와 “개인인증을 도입한다”라는 하위목표가 모순됨
 - 모순은 Goal Tree를 세련화하는 과정에서 해소됨
 - “온라인회원등록”이라는 목표를 선택함으로써 모순 해소 가능성
 - 대표적인 목표지향분석기법
 - I*(아이스타)
 - KAOS

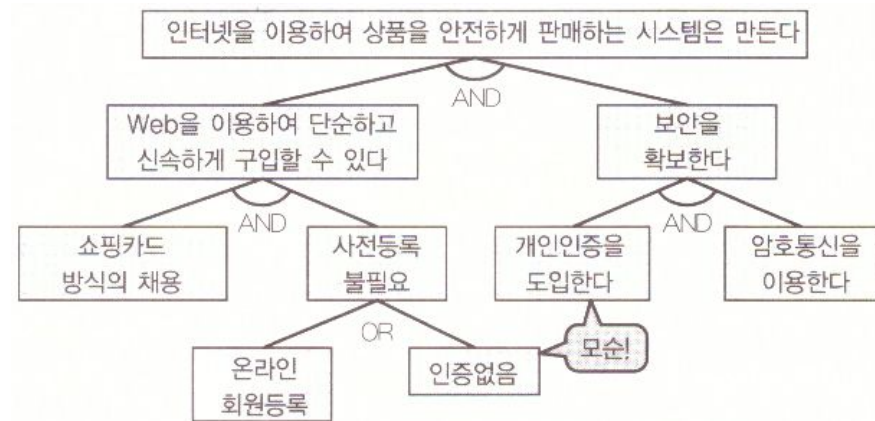


그림 4.2 Goal Tree의 예

4.2.1 요구사항의 추출

- 자주 사용되는 요구사항 추출기법
 - (10) 문제프레임(Problem Frame)
 - 자주 출현되는 문제를 패턴화한 것
 - 현실세계에 존재하는 문제에 대하여 3가지 개념들과 이들사이의 관계에 의해 구조화하는 분석기법
 - 개념 3가지
 - 문제를 해결하는 기계
 - 현실세계를 표현하는 문제영역
 - 문제영역에 대한 동작과 성질을 표현하는 요구사항
 - 문제프레임을 이용하면 기계의 사양과 고객 및 이용자의 요구사항을 명확하게 구별하여 문제 분석 가능
 - 그림 예
 - 기계와 문제영역을 연결하는 직선: 기계가 현실세계에 있는 문제영역과 관계하고 있음→기계에 대한 요구사항(기계의 사양)을 표현
 - 문제영역과 요구사항을 연결하는 점선화살표: 요구사항이 문제영역과 관계하고 있음→고객과 이용자의 요구사항을 표현
 - 예) 요구사항이 현실세계의 모델에 대한 동작 및 성질을 참조하거나 제약사항을 부여하는 것

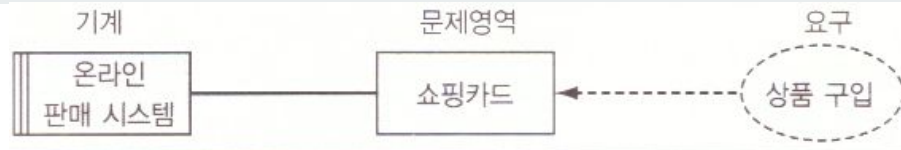


그림 4.3 문제프레임으로 기술한 문제의 예

4.2.2 요구사항의 취사선택

- 요구사항획득에서 서로 모순된 요구사항들이 도출된 경우, 관계자들과 함께 타협점을 찾아내서 요구사항의 우선순위를 결정할 필요가 있음
 - 교섭: 합의형성에 의해 모순을 해소하는 작업
- 요구사항의 트리아지(Triage)
 - 예산과 일정에 의해서 요구사항획득에 의해 얻어진 모든 요구사항들을 만족하는 것이 불가능한 경우 요구사항의 취사선택 필요
 - **트리아지(Triage)**
 - Alan M. Davis 명명
 - 의료용어로서 한정된 의료자원으로 최선의 구명효과를 얻기 위하여 부상자를 부상정도와 긴급성에 따라서 분별하여 치료우선순위를 결정하는 것

4.2.2 요구사항의 취사선택

- 요구사항의 모순을 해소하는 경우, 각 요구사항에 대해서 각각의 관계자들에 의한 점수부여를 수행함으로써 우선순위 결정
- 합의형성기법 이용
 - 델파이법
 - 관계자들이 각자 독자적으로 제출한 의견을 취합하여 상호참조하여 다시 의견을 취합하는 작업 반복
 - 전체적인 경향과 타인의 의견이 피드백됨으로써 전체 의견이 다듬어짐
 - AHP(Analytic Hierarchy Process)기법
 - 계층구조 정의
 - 최상위: 문제
 - 중간: 해결책과 대체안을 선택하는 평가기준
 - 최하위: 문제에대한 해결책과 대체안
 - 각각의 계층에 대해서 해결책과 대체안 평가
 - 최적의 해결책 또는 대체안 결정

개요

- 구조화분석에서의 요구사항 작성절차
 - 현행 물리모델
 - 현행업무의 실태조사 및 현행시스템의 요구사항 기술을 토대로 업무내용을 구체적으로 표현
 - 현행 논리모델
 - 현행물리모델로부터 본질적인 정보만을 남김
 - 관례적으로 수행되고 있는 처리 및 특수한 사정으로 인해 수행되는 처리, 인명, 매체, 일시, 금액, 수량 등 물리적인 제약사항들을 배제
 - 신규 논리모델
 - 현행논리모델에 대해서 새롭게 추가하려는 기능 및 배제하려는 기능을 추출하여 시스템이 가져야할 모습 표현
 - 신규 물리모델
 - 신규논리모델에 대해서 시스템이 만족해야하는 제약사항 및 성능 등을 고려하여 구체화 정리
 - 요구사항 : 신규물리모델을 사양화한 것으로서 비형식적인 도면 표현 및 형식적인 사양 사용

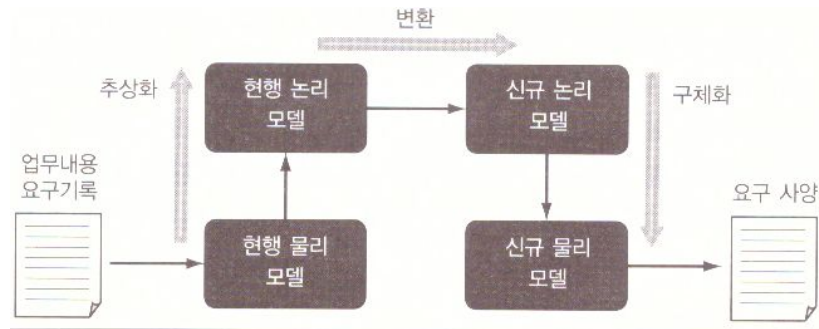


그림 4.4 구조화 분석에서의 요구사항 작성절차

목차

- 4.3.1 도면을 이용한 표현
- 4.3.2 형식적 사양

4.3.1 도면을 이용한 표현

- (1) DFD(Data Flow Diagram, 자료흐름도)
 - 시스템내부의 데이터흐름을 중심으로 업무 및 시스템을 분석하는 구조화 기법에서 사용
- (2) IDEFO
 - 미국 표준화기구인 NIST(National Institute of Standards and Technology)에 의해 공개되어있는 비즈니스모델링기법들의 총칭
 - 업무에서의 활동을 모델링하는 기법
 - 그림 예) 각각의 활동에 대해서 입력, 출력, 제어, 기구(메커니즘)가 정의됨
 - 그림의 업무활동: 주문확인업무는 지불상황에 따라서 주문접수담당자가 주문정보를 입력하고 출력으로서 발송의뢰서를 작성한다
 - 기능을 중심으로 함
- (3) UML
 - 객체지향분석/설계에서 사용되는 통일표기법
- 이외
 - State Transition Diagram (상태전이도)
 - Use Case Diagram 등이 있음

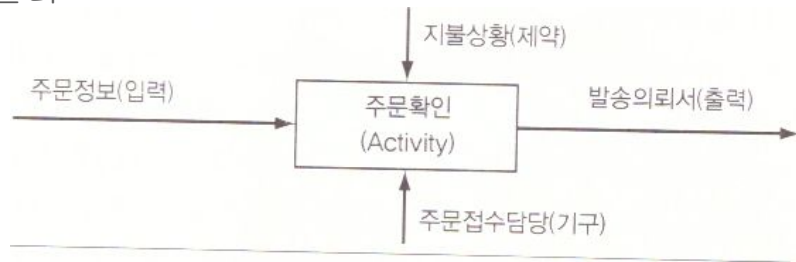


그림 4.5 IDEFO의 표기법

4.3.1 형식적 사양

- 요구사항 정의를 위해 비형식적인 도면표현이나 자연언어가 아니라 보다 **형식적인 언어 및 모델을 사용하여 요구사항을 정의**
 - 자연언어 : 한글, 영어, 일본어 등 우리들이 일상적으로 사용하고 있는 언어
 - 장점: 인간이 다루기 쉬움
 - 단점: 문법이 애매모호하고 표현의 다양성 측면에서 엄밀성이 크게 결여되어 있음
 - 요구사양에 애매모호함이 혼입되기 쉬움
 - 해석을 위한 처리계를 작성하기 어렵기 때문에 기계에 의한 타당성 검증을 지원할 수 없음

4.3.1 형식적 사양

- 형식적 사양
 - 문법과 의미가 형식적으로 정의된 언어와 그림을 사용하여 기술된 사양
 - 장점
 - 누락사항, 애매모호함 등 사양에 오류가 혼입될 가능성 감소
 - 문법과 의미가 수학을 기초로하여 정의되기 때문에 사양을 기계적으로 처리하고 내용을 해석 및 검사할 수 있음
 - 단점(크게 보급되지 않은 이유)
 - 일반적인 개발자와 이용자에게 친숙하지 않음
 - 어느 정도의 수학적 배경을 가지고 있지 않으면 사양을 읽기 어려움
 - 최근 형식적사양 사용 및 계속 보급 예상
 - 컴포넌트지향개발 (CBD: Component-Based Development) 및 모델기동형 개발(MDD: Model Driven Development)에서 OCL(Object Constraint Language) 사용

4.3.1 형식적 사양

- (1) 논리형 사양
 - 시스템의 기능을 사전조건과 사후조건으로 파악하여 논리식으로 기술
 - 사전조건(사전표명): 실행 전에 성립되어야 하는 조건, 입력조건을 기술한 논리식
 - 입력데이터에 관한 제약사항
 - 입력시의 내부상태에 관한 전제조건
 - 사후조건(사후표명): 실행 후에 성립되어야 하는 조건, 출력조건을 기술한 논리식
 - 출력데이터에 관한 제약사항
 - 시스템을 구현한 프로그램을 실행하기 전에 사전조건이 참이라면 프로그램을 실행한 후에 사후조건이 참이 된다는 사실이 보증됨
 - 프로그램의 정당성을 검증하기 위해 이용가능

4.3.1 형식적 사양

- (2) 함수형 사양
 - 시스템의 기능을 입력데이터로부터 출력데이터로의 변환으로 파악하여 직접적인 변환을 수학적 함수에 의해 기술
 - 기능을 함수로 생각하기 때문에 논리형 사양과는 서로 다르며 내부상태를 가정하지 않음
 - 함수가 만족해야하는 동작을 집합 및 등식 등으로 표현
- (3) 대수적 사양
 - 시스템의 기능을 데이터와 그에 대한 연산의 쌍으로 특징지워서 연산의 의미를 대수적 공리(등식)로 기술
 - 추상데이터형에서는 연산의 사양(인터페이스)과 내부구현을 분리하여 공개연산자를 통해서만 데이터에 접근 가능
 - 소프트웨어에서의 추상데이터형 (ADT: Abstract Data Type): 데이터와 이에 대한 연산의 쌍

개요

- 요구사항 확인
 - 작성된 요구사항이 올바른지를 확인하는 작업
- 요구사항의 품질특성(IEEE Std 830-1998)
 - (1) 타당성(정당성)
 - 요구사항에 포함된 모든 요구사항들이 개발하려는 시스템에서 필수적으로 만족해야 하는 사항들인 경우
 - 구현할 필요가 없는 요구사항 불포함
 - 타당성은 고객과 이용자에게 직접 확인받는 것이 일반적
 - 어떤 요구사항이 필수적인지 여부는 고객과 이용자만 판단할 수 있는 경우가 많음
 - (2) 비애매모호성
 - 모든 요구사항이 일의적으로 해석가능한 경우
 - 애매모호한 부분이 존재
 - 설계자와 프로그래머가 요구사항을 잘못 해석하여 최종적으로 이용자와 고객이 의도하지 않은 시스템으로 개발될 가능성이 높아짐
 - 애매모호함은 자연언어로 기술된 요구사항에 혼입되기 쉬움

- 요구사항의 품질특성(IEEE Std 830-1998)
 - (3) 완전성
 - 필요한 정보가 모두 기술되어 있는 경우
 - 기술되어 있지 않으면 안되는 정보
 - 기능, 성능, 설계제약사항 등에 관한 요구사항
 - 요구사항에 등장하는 용어의 정의와 도표에 대한 설명
 - (4) 무모순성(일관성)
 - 요구사항에 기술되어 있는 요구사항들이 서로 모순되지 않음
 - 서로 모순되지 않고 불필요하게 중복되어 있는 요구사항들 중에서 어느 한쪽만 변경되는 경우에 모순이 발생하기 쉬움
 - 요구사항이 불필요하게 중복 : 같은 요구사항이 중복되어 있어서 어느 한쪽을 삭제하더라도 타당성과 완전성을 만족하는 경우를 말함

- 요구사항의 품질특성(IEEE Std 830-1998)
 - (5) 중요도와 안전성에 대한 우선순위부여
 - 모든 요구사항들이 같은 수준의 중요도와 안전성을 요구하고 있지 않음
 - 어떤 요구사항을 우선적으로 다룰 것인가를 결정하기 위한 우선순위 부여
 - 중요도 부여
 - 필수적 요구사항: 반드시 있어야하는 요구사항
 - 조건적 요구사항: 있는 것이 좋은 요구사항
 - 임의적 요구사항: 있어도 좋고 없어도 좋은 요구사항
 - 안전성
 - 요구사항이 어느 정도 변경될 것인지를 나타냄
 - 예) 향후의 변경횟수에 대한 예측값에 의해서 우선순위 부여

- 요구사항의 품질특성(IEEE Std 830-1998)
 - (6) 검증가능성
 - 완성된 시스템이 요구사항의 기술내용을 만족하는지를 체계적으로 검사할 수 있음
 - 애매모호한 표현을 포함하는 요구사항은 검증불가
 - 요구사항에 구체적인 수치가 기술되어 있지만, 타당한 비용(노력과 시간)으로 검사가 완료되지 않은 경우 검증가능하다고 말할 수 없음
 - (7) 변경가능성
 - 요구사항에 기술되어 있는 특정한 요구사항만을 또는 요구사항의 일부분만을 수월하고 완전하며 모순없이 수정가능함
 - 변경가능성이 저하
 - 각각의 요구사항이 독립적이지 않고 서로 의존되어 있는 경우
 - 불필요하게 중복되어있는 경우
 - 특정한 요구사항 및 요구사항의 일부분을 완전하면서도 모순없이 바꾸어서 기술하는 것은 어렵게 됨

- 요구사항의 품질특성(IEEE Std 830-1998)
 - (8) 추적가능성
 - 후방추적이 가능
 - 요구사항의 각 요구사항들에 대해 배경과 이유, 의도가 명확하고 수월하게 참조가능
 - 요구사항을 이해하는 경우 중요
 - 전방추적이 가능
 - 요구사항을 토대로 작성된 설계문서, 소스코드, 매뉴얼 등으로부터 수월하게 참조가능하도록 각 요구사항에 번호와 레이블을 부여
 - 시스템의 유지보수공정에서 설계문서 및 소스코드가 변경되는 경우, 해당 변경의 영향범위를 파악할 때 도움

각 질문에 대한 알맞은 답의 번호(숫자)를 괄호안에 표시하시오.

- ❑ 1. 요구사항분석 작업의 관계자 중 개발하려는 시스템을 필요로 하는 조직 및 기업에서 요구사항의 결정과 개발시스템의 인수작업에 관한 권한을 갖는 사람은 누구인가?
(1) 고객 (2) 이용자 (3) 개발자 (4) 분석가
- ❑ 2. 관계자들을 업무내용 등을 기준으로 여러 그룹들로 나누어서 시스템에 관한 자유로운 의견을 이야기 나눌 수 있는 기회를 제공하고 이러한 대화로부터 요구사항 추출하는 기법은 무엇인가?
(1) 문제프레임 (2) 목표지향분석 (3) 브레인스토밍 (4) 시나리오 (5) 유즈케이스
- ❑ 3. 특정한 상황에서 이용자가 목표를 달성하기 위해 수행하는 행동과 그 행동에 의해 얻어지는 이미지를 시간흐름에 따라서 기술한 것으로서 자연언어, 도면, 애니메이션, 동영상 등으로 표현되는 기법은 무엇인가?
(1) 문제프레임 (2) 목표지향분석 (3) 브레인스토밍 (4) 시나리오 (5) 유즈케이스

객관식 문제 (답안지)

- ❑ 1. (1) → 고객
- ❑ 2. (3) → 브레인스토밍
- ❑ 3. (4) → 시나리오

각 질문에 대한 알맞은 답의 번호(숫자)를 괄호안에 표시하시오.

- ❑ 4. 현실세계에 존재하는 문제에 대하여 문제를 해결하는 기계, 현실세계를 표현하는 문제영역, 문제영역에 대한 동작과 성질을 표현하는 요구사항 등과 같은 3가지 개념들과 이들사이의 관계에 의해 구조화하는 분석기법은 무엇인가?
(1) 문제프레임 (2) 목표지향분석 (3) 브레인스토밍 (4) 시나리오
(5) 유즈케이스
- ❑ 5. 구조화분석에서의 요구사항 작성절차 중 현행업무의 실태조사 및 현행시스템의 요구사항 기술을 토대로 업무내용을 구체적으로 표현한 모델은 무엇인가?
(1) 현행 물리모델 (2) 현행 논리모델 (3) 신규 논리모델 (4) 신규 물리모델
- ❑ 6. 요구사항의 품질특성 중 요구사항에 포함된 모든 요구사항들이 개발하려는 시스템에서 필수적으로 만족해야 하는 사항들인 경우에 해당하는 것은 무엇인가?
(1) 완전성 (2) 타당성(정당성) (3) 비애매모호성 (4) 무모순성(일관성)
(5) 변경가능성

객관식 문제 (답안지)

- ❑ 4. (1) 문제프레임
- ❑ 5. (1) 현행 물리모델
- ❑ 6. (2) 타당성(정당성)

각 질문에 대한 알맞은 답의 번호(숫자)를 괄호안에 표시하시오.

- ❑ 7. 요구사항의 품질특성 중 필요한 정보가 모두 기술되어 있는 경우에 해당하는 것은 무엇인가?
(1) 완전성 (2) 타당성(정당성) (3) 비애매모호성 (4) 무모순성(일관성)
(5) 변경가능성

- ❑ 8. 요구사항의 품질특성 중 요구사항에 기술되어 있는 특정한 요구사항만을 또는 요구사항의 일부분만을 수월하고 완전하며 모순없이 수정가능한 경우에 해당하는 것은 무엇인가?
(1) 완전성 (2) 타당성(정당성) (3) 비애매모호성 (4) 무모순성(일관성)
(5) 변경가능성

객관식 문제 (답안지)

- ❑ 7.(1) 완전성

- ❑ 8.(5) 변경가능성

주관식 문제 (괄호안에 알맞은 단어를 입력하시오.)

- ❑ 1. ()이란 소프트웨어를 이용함으로써 실현하려는 내용을 모아서 정리한 것이고, ()이란 요구사항을 구현하기 위해 필요한 기능과 성능을 제공할 때의 조건 및 제약사항이다.
- ❑ 2. ()이란 개발해야하는 소프트웨어시스템에 대해서 고객과 이용자가 명시적 또는 잠재적으로 생각하고 있는 요구사항을 도출하여 시스템 전체의 사양을 가능한한 엄밀하게 정의하는 작업이다.
- ❑ 3. 요구사양에 기술되어 있는 요구사항은 ()요구사항과 ()요구사항으로 구분된다.
- ❑ 4. ()이란 고객 및 이용자가 진정으로 요구하는 사항을 도출하여 요구사항으로 종합정리하는 작업이다.
- ❑ 5. ()은 시스템을 개발하는 것은 어떤 목표를 달성하기 위해서라는 생각을 토대로 하는 기법이다.

주관식 문제 (답안지)

- ❑ 1. 요구사항, 사양
- ❑ 2. 요구사항분석
- ❑ 3. 기능적, 비기능적
- ❑ 4. 요구사항획득
- ❑ 5. 목표지향분석

주관식 문제 (괄호안에 알맞은 단어를 입력하시오.)

- ❑ 6. 목표지향분석에서 목표는 2종류의 분해에 의해 계층화된다.()
는 분해되는 부분목표가 모두 만족되면 상위의 목표가 만족됨을
나타내고, ()는 부분목표들 중 한 개가 만족되면 상위의 목표가
만족되는 것을 나타낸다.
- ❑ 7. ()란 의료용어로서 한정된 의료자원으로 최선의 구명효과를
얻기 위하여 부상자를 부상정도와 긴급성에 따라서 분별하여
치료우선순위를 결정하는 것이다.
- ❑ 8. ()란 요구사항기술로부터 오류 및 불필요한 부분을
제거하거나, 애매모호한 부분을 배제하거나, 부족한 정보를 보충하여
요구사양을 완성시키는 작업이다.

주관식 문제 (답안지)

- ❑ 6. AND분해, OR분해
- ❑ 7. 트리아지(Triage)
- ❑ 8. 요구사항의 사양화

Any questions?