

중간고사

2020년 5월 3일 일요일 오후 4:30

1단원
Software Object(목표): 하드웨어 동작 작업 관리 수행
소프트웨어란 : programs and associated documentation like design model , requirements
:모든국가 산업 소프트웨어기반
software costs>hardware costs
test(테스트)의 4파라미터로 크다.
software 종류 : custom software
what is software engineering: 소프트웨어 생산에 관련된 모든것들
engineering vs science : 실질적인 개발vs 진리와 이론 관련
system eng" vs software eng -SYSTEM이 다른 의미 architecture hardware 까지 포함

software process란? in Generic. 일반적으로
Specification: 구체화
Development: 개발
Validation: 유효성판단 어떤고객이 원하는지
Evolution: 요구에따른 변화

소프트웨어 모델링 방법론:
waterfall 폭포수:
iterative development(반복적 모형)
component-based software:
development and evolution costs for long-lifetime sync

엔지니어링 방법들 소개:
18페이지 : case Tool 만들어지는데 쓰이는 툴들 디버깅툴,관리툴 등등
19페이지 : what is good software:
Maintainability : 요구에따른 변경할수있어야함
Dependability ,Efficiency Acceptability: 유지율이 높수있어야함

20페이지: 소프트웨어 엔지니어링 어려움:
1)heterogenity: heterogeneous platforms
2)delivery: 기술발전에 따른 빠르게 전달
3)risk
21페이지: 윤리적책임감
22페이지: responsibility(책임감) : Confidentiality(기밀성),Competence(능력)
(intellectual property right (지식재산권), Computer misuse(오용)

3단원
Critical Systems 종류
Safety : 인간의 육체및 환경에 영향
Mission : 해당목적이 상당히 중요할때
Business : 상당히 높은 경제적인 손실
5페이지: critical system에서 가장중요한 건 System dependability, 얼마나 신뢰할수있는지
6페이지: 중요성, 해당시스템이 문제를 일으킬경우 상당히 큰 비용을 일으키기때문에
7페이지: critical system 개발방법, 다른 개발 방법들 이할 다름, critical은 실패비용을 최대한 낮춰야할 개발비용은 중요지않음
8페이지: hardware , software ,Operation(인간의실수) failure
9~11 인술된 관련된 절차=critical system
13,14페이지 dependability 이것들이 만족해야함
1.availability:주어진 시간안에 서비스를 받을수 있을 확률
2.Reliability, 시스템이 정상적으로 돌아가는확률
3.safety
4.security
15페이지: 고려해야할 다른 조건들: Repairability Maintainability(유지보수), Survivability Error tolerance 등등.
18페이지: Dependability vs performance
19페이지: Dependability increase exponentially
21~21페이지:관련설명
23.reliability가 낮아도 무조건 availability하거나 쉽게 repair가능하거나 큰 손해가없다면 상관없다
24.Reliability terminology: System failures 항상 fault,error가 failure아기시키진 않는다.
build-in error detection이 방지시킴,
26. 일반user와 전문가들의 경우 사용확률이 다를 따라서 환경에따른 reliability가 다를
27.reliability를 확보하기 위한 방법들: Fault avoidance, detection and removal ,tolerance 잘 설계해야함
28~31. Reliability models
34. Safety Is reliability 정상작동하지않아도 Safety를 일으키지않아야함
35. unsafe reliable은 Specification errors, Hardware failure
36. Safety용어설명
37.Safety 확보조건 네가지최소화 위험최소화
38. 일반적으로발견시 문제점찾기 쉽지않음
41. security terminology
43 security 확보조건

4단원software process
3page: 기본적인 소프트웨어프로세스
1.Specification :구체화
2.Design and implementation: 디자인 및 구현
3.Validation: 유효성판단
4.Evolution.요구에따른변화

4page:
Process descriptions(절차 명세서):
1.products 2.roles,3.pre and postconditions
5page:
Plan-driven: 선형적으로 계획을 세워놓고 실행
Agile(애자일)프로세스:조금씩조금씩 계획을 세워놓고 점진적으로 실행.
실전에서 두개 다사용함
-----모형
7page:}
소프트웨어 프세스 모델:
1.waterfall(폭포수모형)= Plan-driven 구체화시켜놓고 개발 계획된대로 실행
2.incremental developemts+ 왔다갔다 agile+Plan
3.integration and configuration = 이미 개발된 component를 조합및 수정
8~10. waterfall model: 변경가능성 없을때 쓰이고 상당히 큰모델에서 쓰임 ex)임베디드같은 하드웨어 장치,과정이 잘보임
단점:새로운 요구사항이 있을때 변명하기 힘들
11~12 incremental development: 구체화시킨후 사용자 피드백을 받고 변경하고 개발하고...등
장점: 요구사항 변경 허용, 피드백을 빠르게 받을 수 있음, 사용자(customer)가 빠르게 사용할 수 있음.
단점:specification process 과정이 잘보이진않음.
14 integration and configuration 재사용할수있어서 시간절약가능
16~17
장점: 개발단계가 낮아질수있고 cost가적어지고 risk적어지고 빨라짐.
단점: 사용자의 requirements를 완벽하게 충족못함
소프트웨어 개선및 수정시키기 힘들.
21. requirements engineering process.그림

소프트웨어프로세스
22. specification: requirement 분석및 정립및 constraints를 정립

23~26. design and implementation: Specification 을 디자인및 구현(디버깅포함) 개발시간보다 디버깅시간이 길어짐

27. validation: 고객들의 요구사항에따라 잘 기능여있는지 유효성테스트

29.Testing 단계:
Component testing -> System testing -> Customer testing
(개개인의 component들을 개별적으로 테스트)->System testing

30. testing phase 그림-Vmodels

31,32. Software Evolution: 수정및 변경 -----Changing requirements

34
35: reducing the costs of rework; 변화에대해서 미리 예측 및 변화에 수용할수 있도록 만들어야함
36. 35에양방: system prototyping , increment delivery(애자일식으로 사용자에게 계속적으로 주는것)

=====prototype
37~38. 프로토타입에 대한 설명및 장점 :사용성이좋아지고 quality가 좋아지고 maintainability가 좋아질 유저의 실제필요성충족임
39. 프로토타입::기능에만 초점 또는 ui만 조합됨
40. 애러제강및 requirement포함안시켜도 되고 기능적인 부분제고.
41. 개발한후 prototype은 버려져야함. undocumentation , 구조적으로 안조음

===== increment delivery
42. 개발: 다음 increment전에 현재 increment개발한후 피드백받고 나서 다음 단계로넘어감
43. 진문: 실제 end-users 언급됨
44. 그림
45~46
장점: 1.customer 일찍부터 사용해볼수있고 2.실패확률줄일수있음
3. 고객의 요구에따른 요구사항을 일상하는데 도움을 줄수있다. 4. 테스트를 많이 할수있다.
단점: increment까지는 요인이 구체적으로 규정되지 않기 때문에 모든 increment에서 필요한 공통 설비를 식별하기 어려울 수 있
다.
반복공정의 본질은 소프트웨어와 연계하여 사양을 개발한다. 그러나, 이는 시스템 개발 계약의 일부인 완전한 시스템 사양과 같은 조직의 조달 모델과 상충된다.

5단원Project management
4.
Management activities
Project planning : 일정및 예산산정배치함
Project scheduling
Risk management
1.프로젝트 요구사항에따른 시간제따라 스케줄하기위해서 관리가필요.
2.예산과 채약사항에따라서 소프트웨어를 개발시켜야하기때문에 -----Management activities
6.
Management activities
제안서 작성.

프로젝트 계획 및 스케줄링.
프로젝트 비용.
프로젝트 모니터링 및 검토.
인사 선발 및 평가.
보고서 작성 및 프레젠테이션.

-----project planing:

9.

- 1.가장 시간을 오래삼
- 2.조기의 생각부터 systemdelivery까지 계속해서 계획하여야함.
- 3.다양한 유형의 계획들이 소프트웨어 예산과 스케줄 관련해서 지지하기 위해 발전

제약사항세우고 Risk analysis하고 마일스톤갈장정동용

10페이지 : types of project plan: (example- Quality plan, Validation plan)

11페이지 : planning process

12~13. 프로젝트 plan structure

14: activity organization(활동 조직화)

15.활동-마일스톤 과정

-----18 Scheduling

- *프로젝트를 업무단위로 쪼개고 요구되는 시간과 자원에대해서 예측해야한다.
- *작업 동시에 구성 :최적화
- *지연방지를 위한 작업 종속성 최소화
- *프로젝트 매니저의 직권과 경험에 의존해야한다.
- 17.과정 그림
- 18페이지

문제점: 문제의 난이도와 그에 따른 솔루션 개발 비용을 추정하는 것은 어렵다.

생산성은 어떤 일에 종사하는 사람의 수에 비례하지 않는다.

프로젝트에 사람이 늦게 추가할시 대head때문에 늦게만들어지게된다.

예상치못한문제점발생.

19~23 자트 및 그래프

-----Risk Management

25 Software risk: 직원이 그만둔 매니저변경 하드웨어불가 등등.

26.process: 1.Risk identification(확인) 2.Risk analysis(분석) 3.Risk Planning(계획) 4.Risk monitoring(모니터링)

27그림

28.Risk identification이유:기술,사람,조직적,요구사항,추정동등.

29.Risks and risk types

30~32Risk analysis: 가능성설 및 영향에관한 평가.

33~35Risk planning: Avoidance strategies(피하기위한전략),최소화전략,비상계획

36. 리스크 모니터링: 평가한 위험들이 장기적으로 가능성있는지 평가

위험의 영향이 변하는지 평가

미팅때 중요 위험들은 토론되어야한다.

37.자료

프로젝트 성공을 위해서는 좋은 프로젝트 관리가 필수적이다.

소프트웨어의 무형의 특성은 경영상의 문제를 일으킨다.

매니저는 다양한 역할을 가지고 있지만 가장 중요한 역할을 한다.

관리자들은 다양한 역할을 가지고 있지만

그들의 가장 중요한 활동은 planning, estimate, Scheduling이다.

planning과 estimate은 프로젝트의 과정 내내 계속되는 반복적인 과정이다.

프로젝트 이징표는 경영진에게 공식적인 진행 상황을 제시하여 예측 가능한 상태를 말한다.

프로젝트 스케줄링은 프로젝트를 보여주는 다양한 그래픽 표현 준비와 관련이 있다.

요점

프로젝트 활동, 기간 및 인력을 보여주는 그래픽 표현 리스크 관리

프로젝트에 영향을 미칠 수 있는 리스크를 식별하고

이러한 리스크가 주요 위험으로 발전하지 않도록 하기 위한

계획과 관련이 있다.

-----6단원 Software

Requirements

사용자요구사항: 일반적으로 자연어를 사용자에게 알기쉽게 설명

시스템요구사항:

기능적요구사항:

비기능적요구사항:

domain 요구사항

요구사항.

4~5서비스의 시스템 제약사항에관한 높은단계의 추상성설명부터 자세한 수학적으로 설명하는것까지 포함한다.

계약 입찰의 근거가 될 수 있으므로 해석에 개방적이어야 한다.

*계약 자체의 근거가 될 수 있으므로 세부적으로 정의해야 한다.

*이 두 문장은 모두 요구사항이라고 할 수 있다.

7페이지:

사용자요구사항: 시스템이 제공하는것과 기능적인 제약사항들에 대해 자연어와 다이어그램으로 객체들을 위해 쓰여짐

시스템요구사항: 디테일하게 쓰여져있을 개발자들이나 클라이언트와 개발자간의 요구분석명세서

8~9 user system specification and 누가읽나?

10페이지

기능적요구사항: 특정input에따른 시스템이 상황에따라 어떻게 행동하나?

비기능적요구사항: 시간, 제약, 개발 프로세스의 제약, 표준 등 시스템이 제공하는 서비스나 기능에 대한 제약

11~13 예제

14.요구사항 부정확함에대한 문제제하기 혼돈발생

15. Requirements는 Complete, Consistent필요

16.Non-functional requirement에 대한설명 중요성

17.비기능분류

제품 요구 사항

*납품된 제품이 실행 속도, 신뢰성 등 특정한 방식으로 동작해야 한다는 것을 명시하는 요구 사항

조직 요구사항

*조직 정책 및 절차의 결과인 요구사항(예: 사용된 프로세스 표준, 구현 요구사항 등)

외부 요구사항

*시스템 외부의 요인과 시스템 개발 프로세스(예: 상호운용성 요구사항, 입법 요구사항 등)에서 발생하는 요구사항

18. non functional type

19.예제

20.요구사항

목표:

*사용자의 편의

*검증가능한 비기능 요구사항: 객관적으로 시험할 수 있는 몇 가지 수단을 이용한 진술

*이러한것들은 시스템 사용자의 의도를 개발자에게 전달하는데 도움됨

21.예제

22. Requirements measures

23 Requirements interaction

24~26 Domain requirements

-----<도메인 요구사항 (Domain requirements)>

종종 도메인 으로부터 생기는 요구사항으로 특정 도메인의 용어나 개념이 반영되어 시스템의 특성에 영향을 준다.

세로운 기능적 요구사항 또는 기존의 요구사항에 대한 제약사항이 되거나 특별한 계산식을 정의하는 것 등이 될 수 있다.

도메인의 요구사항이 만족되지 않으면 시스템이 잘못 설계 된다.

도메인: 해당 응용 도메인어 언어로 표현되기 때문에 시스템을 개발하는 소프트웨어 공학자가 이해하기 어렵다.

도메인 전문가에게는 흔명한 부분이기 때문에 생략해버린 정보를 개발자가 알 수 있다.

-----User requirement(사용자요구사항)

28~29 특징: 기능적+비기능적인 요구사항을 지식이 깊지않은 시스템사용자가 이해할수있어야함+자연어

문제점:명확성부족+혼동+충돌

30~32 requirement 해석및 문제점

33 내용

34.요구사항 가이드라인:

표준 형식을 만들어 모든 요구 사항에 사용하십시오.

언어를 일관성 있게 사용하라.

필수 요건에 사용되어야 하며, 바람직한 경우 사용되어야 한다.

텍스트 강조 표시를 사용하여 요구 사항의 주요 부분을 식별하십시오.

런타임 용어의 사용을 피하십시오.

-----System requirements(시스템 요구사항)

35. 시스템기능과 유지요구사항의 제약서보다더 디테일해야함

system의 설계를 intended해야함

-시스템계약포함 8장이상연관

-----36. Requirements and design

-----Language specifications

37.NL specification 문제점 :Ambiguity ,Over-flexibility,Lack of modularisation(모듈화)

38.대안 structure language or design or Graphical 등등등.

39 structured language 특징:자연어의 표현력은 대부분 유지되지만 규칙에 교집합이 부여되는 것이 장점이다.

40. Form-based specifications특징 :Definition of the function or entity. 함수와 entity정의, 입력과 출력 등등 표현

input output 등등이주어짐

41.Tabular specification특징 :자연어를 보충하기 위해 사용된다. 가능한 여러 가지 대안적 행동 과정을 정의해야 할 때 특히 유용하

다.

condition 과 Action이주어짐

44.graphical models. 어떻게 상태가 변하는지 또는 단계설명할때 가장 효과적

45.46. Sequence diagrams: 상호작용에따른 순서가 어떻게 일어나는지 설명할때중용

-----Interface specification

47~48대부분의 시스템은 다른 시스템과 함께 작동해야 하며 작동 인터페이스는 요구사항의 일부로 명시되어야 한다.

세 가지 유형의 인터페이스를 정의해야 할 수 있음

결과 인터페이스

교환되는 데이터 구조

데이터 표현

공식 형식은 인터페이스 사양에 효과적인 기술이다. 예시

-----The requirements document

49~50요구사항 문서는 시스템 개발자에게 요구되는 사항에 대한 공식 서명이다.

사용자 요구사항의 정의와 시스템 요구사항의 사양을 모두 포함해야 한다.

그것은 디자인 문서가 아니다. 가능한 한 how가 아니라 시스템이 무엇을 해야 하는지를 설명해야 한다.

그림 사용자 요구사항 문서 표시

51 -IEEE requirements standard :

1. introduction(소개)

2. General description (일반화)

3. Specific requirements(요구사항구체화)

4. Appendices(부록)

5. index(인덱스)

-----Requirements document structure

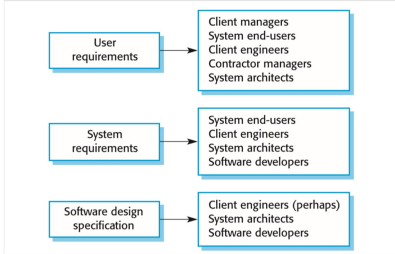
2. 요구사항 문서구조

: 시스템 requirements 은 시스템이 제공해야 하는 기능을 전달하기 위한 것이다.

소프트웨어 요구사항 문서는 시스템 요구사항에 대한 합의된 진술이다.

IEEE 표준은 보다 상세한 특정 요구사항 표준을 정의하기 위한 유용한 출발점이다

Requirements readers



주요 요구사항 엔지니어링 활동 및 관계 설명

요구사항 도출 및 분석을 위한 기법 도입

요구사항 검증 및 요구사항 검토의 역할 설명

다른 요구사항 엔지니어링 프로세스를 지원하기 위한 요구사항 관리의 역할에 대해 논의

3. Topic
Feasibility studies 타당성체크
Requirements elicitation and analysis 요구사항 추출 및 분석
Requirements validation 요구사항 검증
Requirements management 요구사항 관리

4.RE process domain에 따라 다르지만 일반적으로 이러한
Requirements elicitation->Requirements analysis->Requirements validation->Requirements management
5~6과정 그림

7.특징 :보람이 있는 일 인지 아닌지 판단
1)시스템이 조직목표에 기여하는지
2)시스템이 현재 예산안의 기술에서 처리될수 있는지
3)다른시스템과 충돌될 수 있는지

8.질문 1)문제점이유니나,새로운기술필요하나 등등

-----Elicitation and analysis

9
특정, 고객과 협력하여 애플리케이션 도메인 시스템이 제공해야 하는 서비스 및 시스템의 운영 행위를 파악한다.
요구사항 도출 및 분석은 도메인 이해, 요구사항 수집, 분류, 구조화, 우선순위화 및 검증과 관련된 반복적인 것이다.
10.문제점, 요구사항은 분석 과정에서 변경된다.
Stakeholders don't know what they really want
새로운 이해관계자가 등장하고 비즈니스 환경이 변화할 수 있다.
조직적, 정치적 요인은 시스템 요인에 영향을 미칠 수 있다.

-----The requirements spiral

11.그림
12.
Requirement discovery:제안된 시스템과 기존 시스템에 대한 정보를 수집하고 사용자 및 시스템 요구사항을 이 정보로부터 확장하는 프로세스
정보 출처에는 문서, 시스템 이해관계자 및 유사한 시스템의 사양이 포함된다.
classification and organisation
Prioritisation and negotiation 선행요구사항 해결요구사항 충돌
Requirements documentation 요구사항 문서
14. ATM stakeholders 예

-----viewpoint

15~16 viewpoint and type 특징Stakeholders에 따라 다른관점이다.
17Identify view pint
다음은 사용자와 관점 식별
•시스템 서비스의 제공자 및 수신자
•지정되는 시스템과 직접 상호작용하는 시스템,•규제 및 표준
•업무 및 비기능 요구사항의 원천
•시스템을 개발하고 유지해야 하는 엔지니어
•마케팅 및 기타 비즈니스 관점
18LBSVS viewpoint hierarchy

-----interviewing

19.
이해관계자한테 시스템관련해서 물어보는것
•사전 정의된 질문 세트가 답변되는 비공개 인터뷰
•사전 정의된 연관이 없고 이해관계자와 함께 다양한 이슈를 탐색하는 공개 인터뷰
20
21요과적인터뷰, 이해관계자의말 경청해야함
22~24 시나리오

-----use-case

25~29
-----Social and organisational factors

-----Requirements validation

35
요구사항이 고객이 진정으로 원하는 시스템을 정의한다는 것을 입증하는 것에 대해 고려한다.
요구사항 오류 비용이 높으므로 검증이 매우 중요하다
•제공 후 요구사항 오류를 수정하는 경우 구현 오류 수정 비용의 최대 100배까지 비용이 소요될 수 있다.

Requirements checking

Validity: :고객들의 needs에 따라 잘 지원

Consistency:충돌하지않는지

Completeness: 고객요구하는모든기능포함?

Realism: :예산안에서 실행가능한지

Verifiability:검증가능성

37.Requirements validation technique: 프로토타입링 리뷰, 테케
38~39 리뷰,리뷰체크

-----Requirements management

특징,
요구사항 관리는 요구사항 엔지니어링 프로세스 및 시스템 개발 중에 변화하는 요구사항을 관리하는 과정이다.
요구사항은 필연적으로 불완전하고 일관성이 없다.
•비즈니스 요구사항이 변화하고 시스템에 대한 더 나은 이해가 개발됨에 따라 프로세스 중에 새로운 요구사항이 나타난다.
•다른 관점은 요구사항이 다르며 이는 종종 모순된다.
41요구사항 변화 관리
42 요구사항 evolution-그림
45.Requirements management planning
요구사항 식별
•요건이 개발적으로 식별되는 방법
변경 관리 프로세스
•요건 변경 분석 시 준수되는 프로세스
추적가능성 정책
•요건 관계에 대한 정보가 유지되는 양
CASE 툴 지원
•요건의 변화를 관리하는 데 필요한 도구 지원

-----Traceability

47matrix

-----Case tool suport

48.
요구 사항 저장
•요구사항은 안전한 관리형 데이터 저장소에서 관리되어야 한다.
변경관리
•변경관리 프로세스는 단계를 정의할 수 있고 이들 단계 간의 정보 흐름이 부분적으로 자동화된 워크플로우 프로세스다.
추적관리
•요건 간의 링크 자동 검색

-----Requirements change management

49.
요건에 대해 제안된 모든 변경사항에 적용되어야 한다.
주요 단계
•문제 분석 요구사항 문제를 논의하고 변경을 제안한다.
•분석 및 비용 변경 기타 요구사항에 대한 변경 효과 평가
•실행 변경 요구사항 문서 및 기타 문서를 변경사항을 반영하도록 수정하십시오.
50.변경관리 그림

Key points
요구사항 엔지니어링 프로세스에는 타당성 조사, 요구사항 도출 및 분석, 요구사항 규격 및 요구사항 관리가 포함된다.
요구사항 도출 및 분석은 도메인 이해, 요구사항 수집, 분류, 구조화, 우선순위화 및 검증과 관련된 반복적인 것이다.
시스템에 서로 다른 요구 사항을 가진 여러 이해관계자가 있을
조직적 및 조직적 요인은 시스템 요인에 영향을 미친다. 요구사항 검증은 유용성, 일관성, 완전성, 현실성 및 검증가능성에 대한 정
검과 관련이 있다.
사업상의 변화는 필연적으로 요구사항의 변화로 이어진다. 요구사항 관리에는 계획 및 변경 관리가 포함된다.

강 시스템모델링

2.

Context Models:

Interaction Models

Structal Models

Behavioral models

3.System modeling이란: 시스템을 추상적(구체적(x))인 모델로 단순화하여 만드는 단계
다양한 viewpoint에 따른 모델링(UML(Unified modeling Language))

4.이미존제하는 계획된 모델들은 새로운 시스템의 requirements이들어내는데 쓰임
requirement engineering에 쓰임

5. System perspective
An external perspective, 시스템의 컨텍스트 또는 환경을 모델링하십시오
An interaction perspective, 여기서 시스템과 해당 환경 또는 시스템 구성 요소 간의 상호작용을 모델링하십시오.
A structural perspective, 여기서 시스템의 구성 또는 시스템에서 처리되는 데이터의 구조를 모델링한다
A behavioral perspective동적으로 변하는 행동에따라서 이벤트에따라서

6.UML(Unified model Language)객체지향 모델링의 표준, 설명
Activity diagrams:행동이 포함하고 있는 프로세스 또는 데이터접근
프로세스의 액티비티들과 한 액티비티에서 다른 액티비티로 제어의 흐름을 보여줌
Use case: 시스템과 환경(사용자 다른시스템)사이에서 상호교환
Sequence : 시스템이 동작할때 sequence중심으로 Actor와 System componet와 어떻게 상호
Class: 객체클래스중심으로 바라보고 Object와 Class를중심으로 보여줌
State: system이 내부,외부에서 어떻게 반응하는지 보여줌
7.참고로~

-----Context models

9~12시스템의 경계에 있는게 뭔지 알려줌 주변환경과 상호작용을 어떻게하는지
주어진환경과 다른환경과 어떤관계에있는지 보여주고 상호작용은 자세히보여주는것은않음
문제점:속도,동작부위,의존성

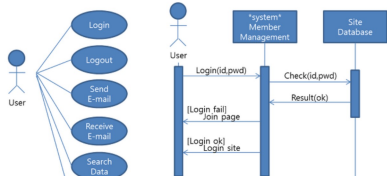
-----Process model

13page :시스템외부와 시스템이 어떻게 연결 Activity diagram

-----Interaction Model(사용자<->

SYSTEM SYSTEM <-> SYSTEM <-> SYSTEM <-> SYSTEM 사용자 <->

6.3 사용자 인터페이스의 Use Case와 Login Sequence Diagram



주어진 문제에서 어떻게 문제를 해결할 것인가에 대한 문제를 T로 나타내며 상호작용을 어떻게 하는지를 나타내며

주어진 환경과 다른 환경과 어떤 관계에 있는지 보여주고 상호작용은 자세히 보여주지는 않음 문제점: 속도, 동작 여부, 의존성

13page: 시스템외부와 시스템이 어떻게 연결 Activity diagram

14~15: 사용자 요구사항을 식별하는 데 도움이 되기 때문에 중요하다.

모델링 시스템과 시스템 간 상호작용은 발생할 수 있는 통신 문제를 강조한다.

모델링 구성요소 상호작용은 제한된 시스템 구조가 필요한 시스템 성능과 신뢰도를 제공할 가능성이 있는지 여부를 이해하는 데 도움이 된다.

Use case diagrams과 sequence diagrams은 상호작용 모델링에 사용할 수 있다.

16~18: 사용자가 어떤 인터페이스에서 어떤 결과 기대하는지 상호작용을 단순하게 기술

requirement가 무엇인지 도출하는데 쓰임

actor로 표현(그림으로)

복잡한내용은 표를 이용해서

20~21

UML의 일부이자 USERCASE에서 발생하는 interaction들에서 사건들을 다룸

가장상단 객체와 actor준제

24.Structural model: Component와 그들의 관계구성 조직적으로 보여줌 1대1 1 1대N

정적인 모델들에서는 시스템의 디자인을 보여주고

동작면모델들에서는 실행할때 어떻게 동작하는지 보여줌

25.객체지향적 관점에서 바라볼때 다이어그램

클래스사이의 연결

논에보이는 모델들가지고 표현

26 그림 Association관계 1대1관계 1대N관계

29.일반화시키기: 시스템 모델링 할때 유리함

1class검사하거나 할때 관련되어있는것만 확인

2상속관계를 잘알수있다.

37. 어떤 data.Event에대해서 어떻게 Dynamic하게 바뀌는지,어떻게 반응하는지

시스템이 환경의 자극(데이터, 이벤트)에 반응할 때 무엇이 일어나는지 또는

무엇이 일어나도록 의도 되었는지를 보여줌

Data type:비즈니스적모델보여주기에 적합

Events type:실시간 모델보여주기적합

38. 데이터가 input오면 어떻게 sequence하게 수행되는지 output이되는지

input->output이러한시스템에 requirement분석하기가 쉽다. 한쪽으론만 흘러감

39 processing모델, 40sequence모델

시스템의상태가 다른상태로 전이됨

State 도식화

이해관계당사자들

Context Models

시스템 상세 초기단계

- 개발되는 시스템의 내용 및 시스템의 경계를 정함

- 시스템 이해당사자들과 함께 작업

- 시스템 비용, 요구사항 및 설계를 이해하는데 필요한 시간을 제한하기 위하여 빠른 결정 필요

비즈니스 프로세스 모델

- 특정 소프트웨어 시스템이 사용되는 자동화된 프로세스들과 사람들을 설명

- Activity Diagram

프로세스의 액티비티들과 한 액티비티에서 다른 액티비티로 제어의 흐름을 보여줌

Interaction Models

Use case modeling

- 시스템과 외부 에이전트(사용자나 다른 시스템들)와의 상호작용을 모델링

Sequence diagram

- 액티와 시스템의 객체들 간의 상호작용과 객체들 간의 상호작용을 모델링

- 외부 에이전트도 상호작용에 포함될 수 있음

구조모델(Structural Models)

- 시스템을 구성하는 컴포넌트들과 그들 간의 관계를 보여줌

동작모델(Behavioral Models)

- 시스템이 실행될 때의 동작 행동에 대한 모델

- 시스템이 환경의 자극(데이터, 이벤트)에 반응할 때 무엇이 일어나는지 또는

무엇이 일어나도록 의도 되었는지를 보여줌

