요구 공학 프로세스

- 시스템 요구사항을 찾아내고, 분석하고, 확인하기 위해 사용되는 프로세스


개요

- 요구 공학의 주요 활동들에 대해 설명함.
- 요구사항 유도(elicitation)와 분석 기법을 소개.
- 요구사항 확인(validation)에 대한 설명.
- 다른 요구 공학 프로세스를 지원하기 위한 요구사항 관리의 역할을 논의함.

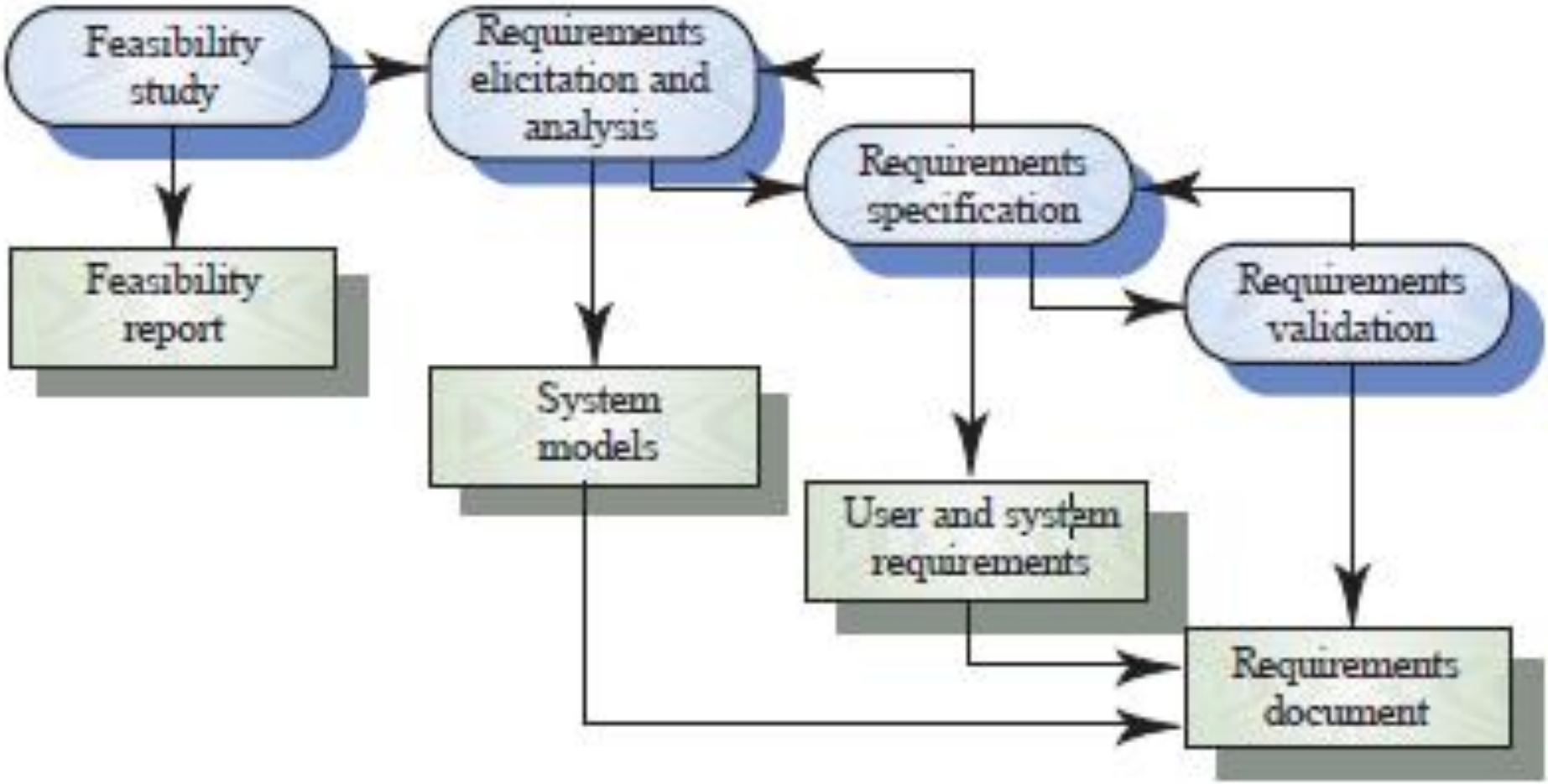
다루는 주제

- 가능성 분석(Feasibility studies)
- 요구사항 유도와 분석
- 요구사항 확인(validation)
- 요구사항 관리


요구공학 프로세스

- RE에 사용되는 프로세스는 어플리케이션의 도메인, 요구사항에 관련된 사람과 개발하는 조직에 따라 상당히 다름.
- 그러나, 모든 프로세스에 공통적인 일반적인 활동들이 있음.
  - 요구사항 유도(elicitation)
  - 요구사항 분석(analysis)
  - 요구사항 확인(validation)
  - 요구사항 관리(management)

요구 공학 프로세스


Feasibility study
Requirements elicitation and analysis
Requirements specification
Requirements validation
Feasibility report
System models
User and system requirements
Requirements document


가능성 분석(Feasibility studies)

- 가능성 분석은 제안된 시스템이 수행할 만한 가치가 있는 것인지를 판단함.
- 다음과 같은 것을 검사함:
  - 시스템이 조직의 목적에 기여할 수 있는가
  - 시스템이 현재의 기술과 가능한 예산 범위 내에서 개발될 수 있는가
  - 시스템이 현재 사용되고 있는 다른 시스템들과 통합될 수 있는가


가능성 분석(계속)

- 정보 수집과 정보에 대한 평가, 그리고 보고서 작성 등의 작업을 수행함.
- 조직내의 인원들에게 묻는 질문들
  - 시스템이 개발되지 않으면 어떻게 되는가?
  - 현재의 프로세스의 문제점들은 무엇인가?
  - 제안된 시스템은 어떻게 도움을 줄 수 있는가?
  - 통합에 관계된 문제점들은 무엇이 있을 수 있는가?
  - 새로운 기술 또는 기법들이 필요한가?
  - 제안된 시스템은 어떤 기능들을 지원하는가?

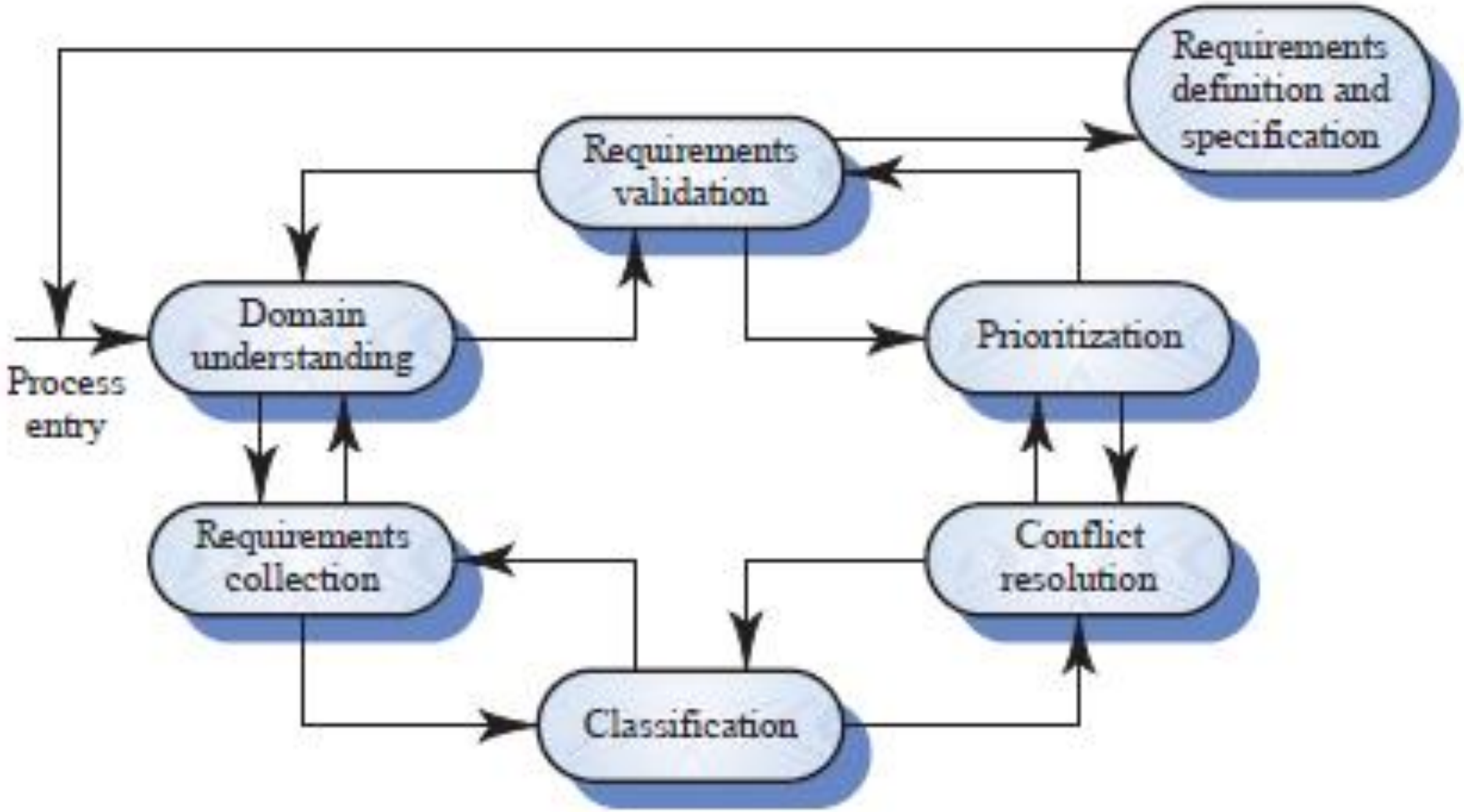
요구사항 유도와 분석

- 요구사항 유도 또는 요구사항 발견이라 불림.
- 기술 직원이 고객과 함께 어플리케이션
- 도메인, 시스템이 제공해야 할 서비스, 그리고 시스템의 작동에 대한 제한사항 등을 찾아냄.
- 최종 사용자, 관리자, 유지보수에 관계된 공학자, 도메인 전문가, 마케팅 관계자 등이 관련됨. 이들을 ***stakeholders***라 부름.


요구사항 분석의 문제점

- Stakeholders들은 그들이 정말로 무엇을 원하는지를 모름
- Stakeholders들은 요구사항을 자신들의 용어로 표현
- 서로 다른 stakeholders 들이 요구사항에 의견 차이가 있을 수 있음.
- 조직의 또는 정책적 요인들이 요구사항에 영향을 미칠 수 있음
- 요구사항이 분석 과정 동안 변함. 새로운 요구사항을 가진 새 stakeholders가 나타날 수 있음

요구사항 분석 프로세스


Requirements definition and specification
Requirements validation
Domain understanding
Prioritization
Process entry
Requirements collection
Conflict resolution
Classification

프로세스 활동들

- 도메인 이해
- 요구사항 수집
- 요구사항 분류
- 요구사항 충돌 해결
- 요구사항 우선 순위화
- 요구사항 검사

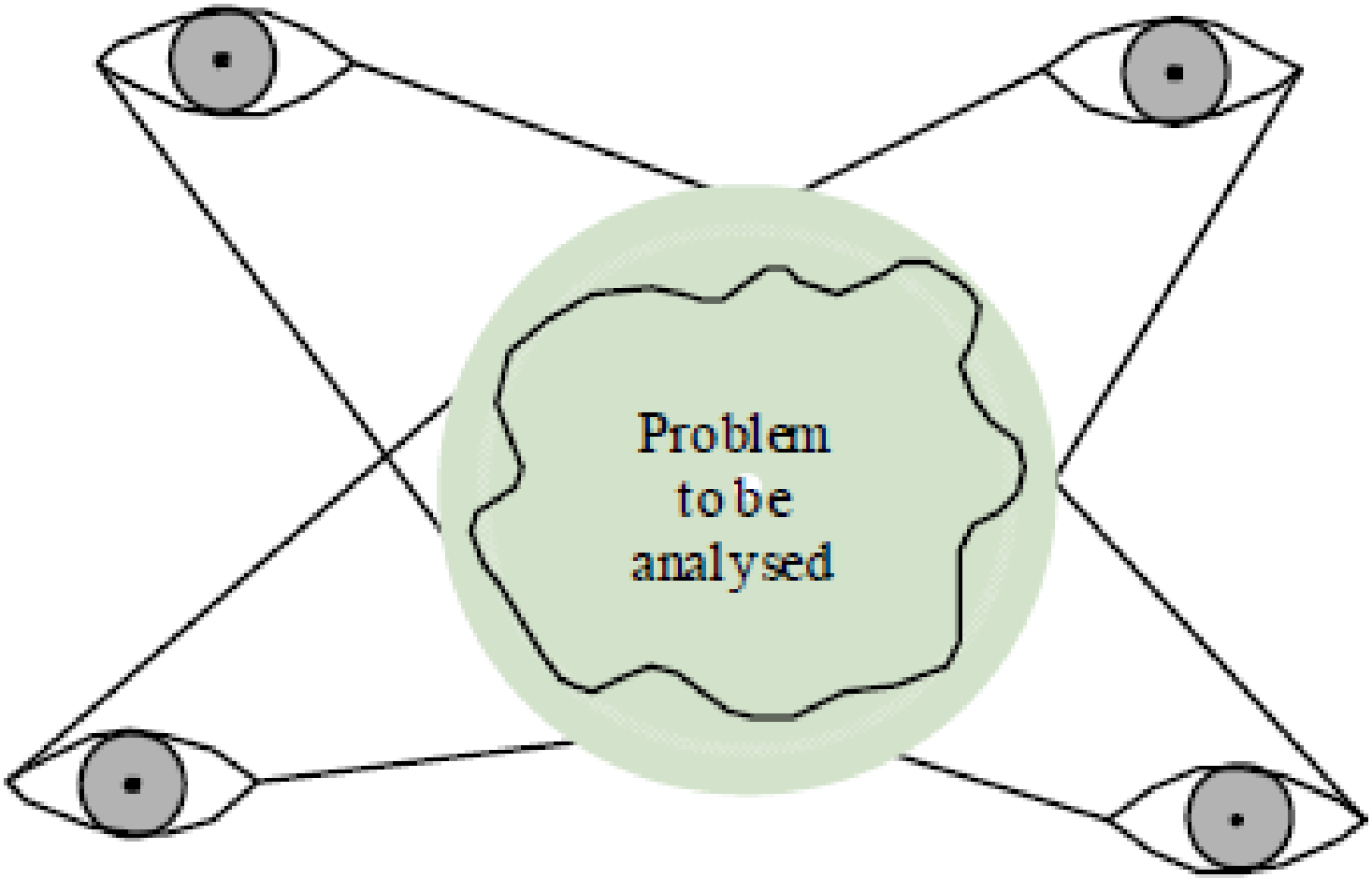

시스템 모델

- 서로 다른 모델들이 요구사항 분석 행위동안 생성될 수 있음
- 요구사항 분석은 3개의 중요한 구조적 작업을 포함.
  - Partitioning: 개체들사이의 구조적 (part-of) 관계를 인식, 요구사항을 분할
  - Abstraction: 개체들사이의 일반성(generalities)을 인식
- • 투사(Projection):문제를 바라보는 여러 관점을 인식
- 7장(시스템 모델)에서 다룸.

관점 중심의 유도

- Stakeholders 들은 문제를 보는 관점이 다름
- 시스템 요구사항을 분석하는 하나의 정확한 방법이 없으므로 여러 측면에서의 분석이 중요함.

Multiple problem viewpoints




현금자동지급기 시스템에 대한 관점

- 은행 고객
- 다른 은행의 대표자 - 금융전산망(네트워크)
- H/W 와 S/W의 유지보수자
- 은행의 판매부 - maintain & upgrade 시스템을 다른 은행에 판매 & 고객 편의를 통한 marketing
- 은행의 시스템 관리자 - 통합
- DB 관리자, 은행의 보안 관리자 - concurrent access
- 통신 관리자 - network
- 은행의 인사부서 - 감원, 고용조정


관점의 유형

- Data sources or sinks
  - 자료의 생산과 소비에 중점을 둠.
- Representation frameworks
  - 관점을 시스템 모델의 특별한 유형으로 생각. 다른 engineer 들이 하나의 시스템에 대해 분석하여 다른 모델(예를 들면, ER, STD, ...)을 개발함.
    - 한 모델에서 missing한 요구사항을 찾을 수 있음
- Receivers of services
  - 시스템의 외부에 있는 관점
  - interactive system에 적합


외부에서의 관점

- 시스템 서비스들의 수혜자인 최종 사용자들의 입장에서는 자연스러운 관점임.
- 요구사항 유도 작업을 구조화할 수 있음.
- 관점이 올바른지를 검사하는 것이 상대적으로 쉬움.
- 비기능적 요구사항들을 구조화하는데 사용될 수 있음.

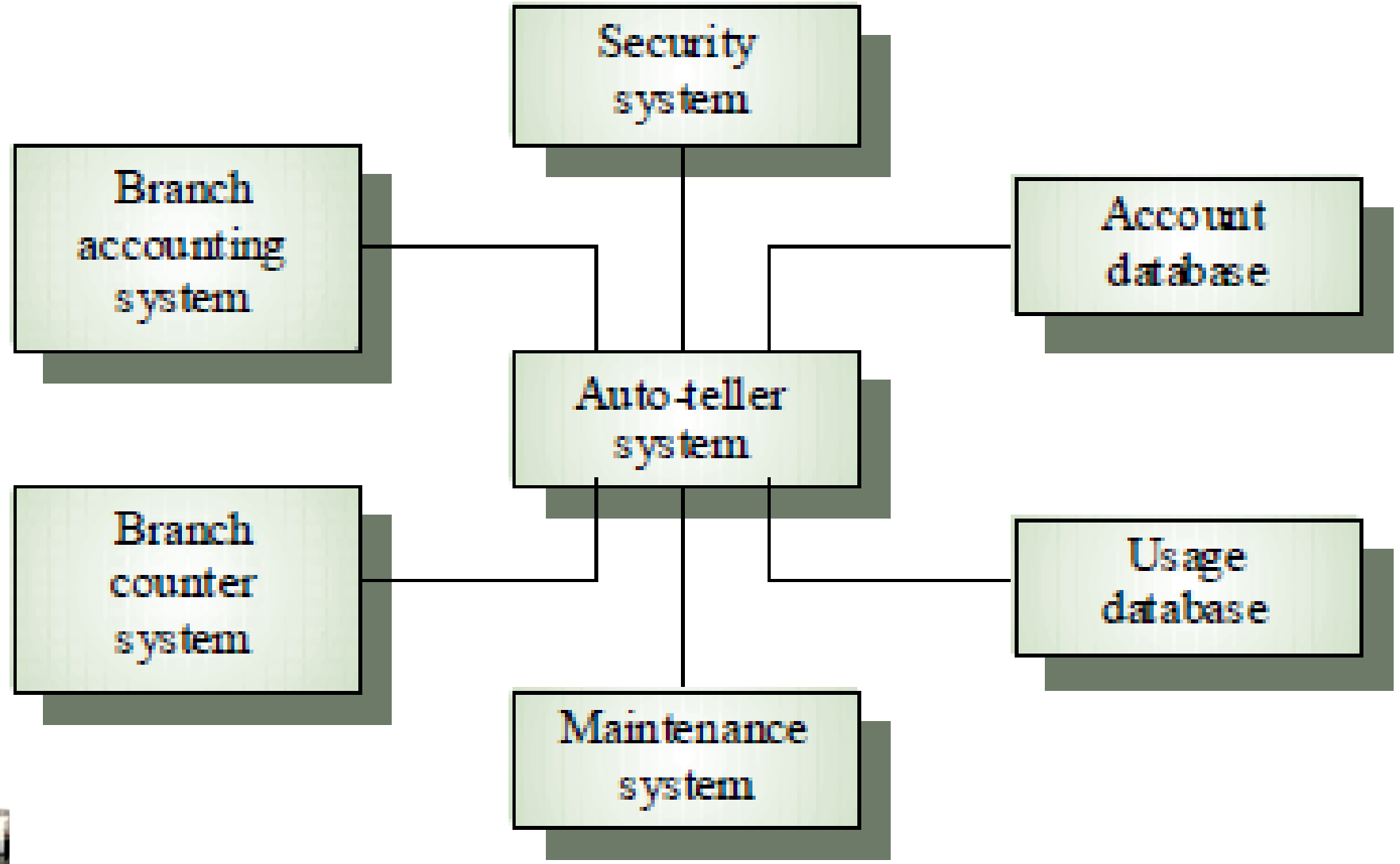
방법 중심 분석

- 요구사항 분석에서 가장 널리 사용되는 방법.
- 시스템을 이해하기 위해 사용되는 방법에 의존.
- 프로세스 모델 수행 방법의 활동들을 정의
- 시스템 모델링 표기법 (DFD, ERD, OSD, Use Case Diagram, ...)
- 시스템 모델에 적용되는 규칙
- 설계 지침
- Report 양식(templates)

System 범위(contexts)

- 무엇이 구현되어야 하는지를 결정하기 위해 시스템의 경계가 설정되어야 함.
- 환경 내에 있는 다른 시스템에 대한 기술도 포함.

Auto-teller system context




Scenarios

- 시나리오는 시스템이 실제로 사용되는 방식에 대한 설명임.
- 이것은 요구사항 유도에 많은 도움을 줌. 왜냐하면, 사람들이 시스템으로부터 무엇을 필요로 하는지에 대해 추상적인 문장들보다 시나리오에 좀 더 쉽게 접근할 수 있기 때문임.
- 시나리오는 개략적인 요구사항 설명에 상세한 것을 추가하는데 매우 유용함.

시나리오 기술 사항

- 시나리오 시작 시의 시스템 상태
- 시나리오에 있는 이벤트들의 정상적인 흐름.
- 잘못된 경우와 이에 대한 대처 방법
- 다른 병행적인 활동들
- 시나리오 종료 시의 시스템 상태


이벤트 시나리오

- '트랜잭션 시작'과 같은 어떤 특정한 이벤트 발생에 대해 시스템이 어떻게 반응하는 가를 설명하는데 사용됨.
- 이벤트 시나리오에 사용되는 다이아그램 규칙 (VORD 의경우)
  - 제공되고 전달되는 데이터
  - 제어정보
  - 예외 처리
  - 예상되는 다음이벤트

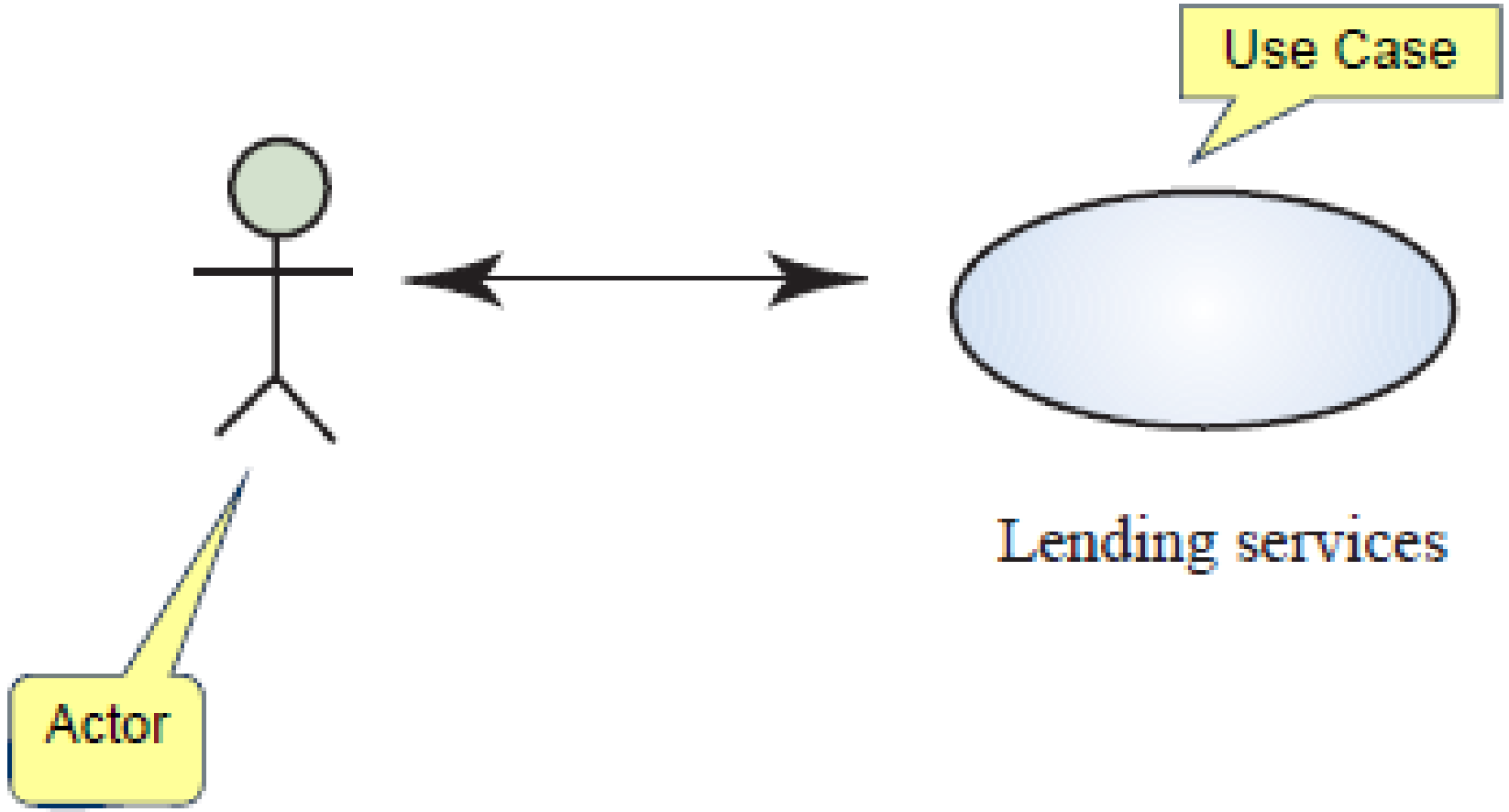
데이터와 제어 분석을 위한 표기법

- 타원. 관점으로부터 제공되거나 또는 전달되는 데이터
- 제어 정보의 입출력은 상자의 윗 부분에 표시
- 데이터는 상자의 오른쪽으로 나감.
- 예외 사항들은 상자의 아래 쪽에 나타냄.
- 다음 이벤트의 이름은 진하게 표시된 상자에 있음.

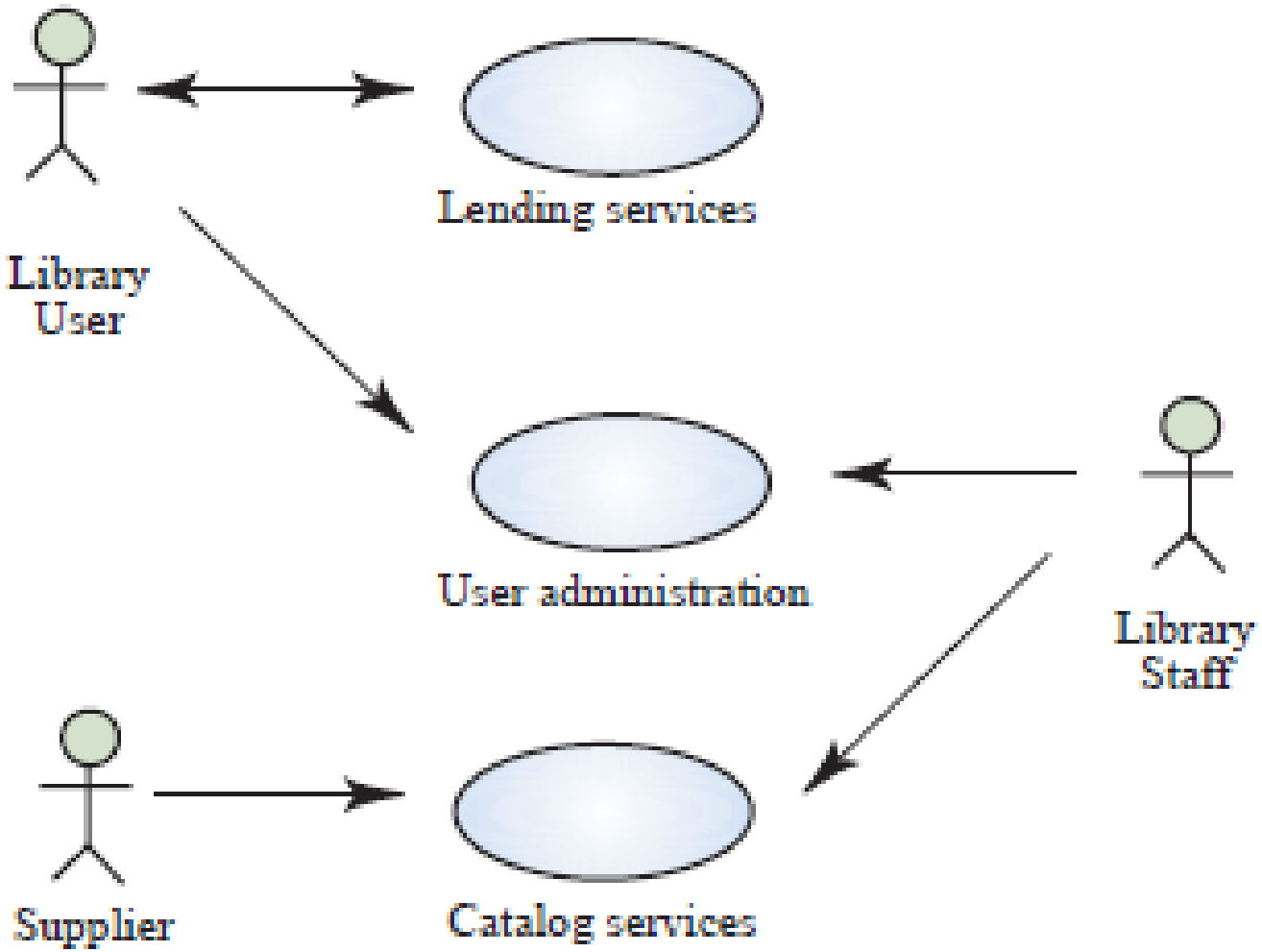

Use cases

- Use-cases는 UML에서 사용되는 시나리오 기반의 기법으로, 시스템에 관련된 외부 행위자(actor)를 표시하고, 외부 행위자와 시스템과의 상호작용에 대해 표현함.
- use cases 집합은 시스템이 갖고 있는 모든 가능한 상호작용을 기술하여야 함.
- 시스템에서의 이벤트 처리 순서를 보여주는 순차 다이아그램(Sequence diagrams)이 use case에 상세한 설명을 추가하기 위해 사용될 수 있음.

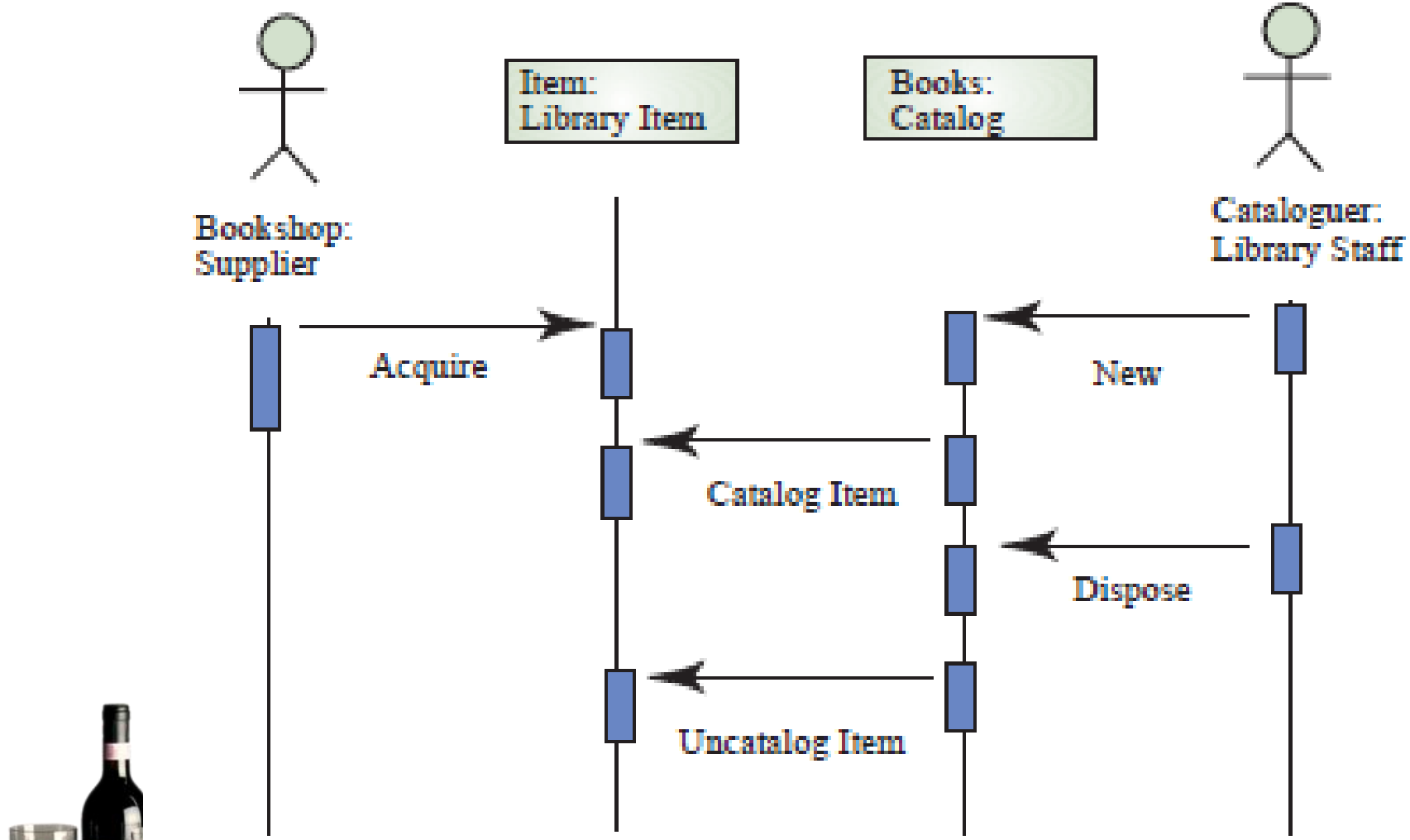
대출(Lending) use-case



도서관 시스템에 대한 usecases



Catalogue management-Sequence diagram



요구사항 확인

- 요구사항이 고객이 정말로 원하는 시스템을 정의했는지를 확인하는 것.
- 요구사항 에러에 따른 비용은 매우 크기때문에 확인은 매우 중요함.
  - 인도 후에 요구사항 에러를 고치는 비용은 구현 에러를 고치는 비용의 거의 100배에 달함.


요구사항 검사

- 정확성(Validity): 시스템이 고객의 요구사항을 가장 잘 지원하는 기능을 제공하는가 ?
- 일관성(Consistency): 요구사항들간에 불일치는
- 없는가?
- 완전성(Completeness): 고객이 필요로 하는 모든 기능이 포함되었는가 ?
- 현실성(Realism): 요구사항이 주어진 가용한 예산과 기술로서 구현될 수 있는가 ?
- 검증성(Verifiability). 요구사항이 검사될 수 있는가?


요구사항 변경

- 개발 프로세스 동안 여러 관점의 변화에 따른 요구사항의 우선 순위 변화
- 시스템 고객들은 사업의 관점에서 최종 사용자의 요구사항과 충돌하는 요구사항을 명시할 수 있음.
- 시스템 개발동안 사업 환경과 기술적 환경의 변화


지속적 요구사항과 휘발성 요구사항

- 지속적 요구사항. 고객 조직의 핵심적 활동에서 나온 안정적인 요구사항
  - (예) 병원에는 항상 의사, 간호사 등이 있다.
  - 도메인 모델로부터 유도될 수 있음.
- 휘발성 요구사항. 개발하는 동안 또는 시스템을 사용하는 동안 변경되는 요구사항
  - (예) 병원에서, 건강 관리 정책으로부터 유도된 요구사항.


요구사항 관리 계획

❖요구 공학 프로세스동안 수립할 계획들:

- 요구사항 인식
  - 요구사항을 어떻게 식별할 것인가.
- 변경 관리 프로세스
  - 요구사항 변경을 분석할 때의 프로세스에 대한 계획
- 추적성(Traceability) 정책
  - 요구사항 사이의 관계를 관리하기 위한 계획
- CASE 도구 지원
  - 요구사항 변경을 관리하는 것을 지원하는 도구들에 대한 계획


추적성(Traceability)

- 추적성은 요구사항, 요구사항의 근원, 그리고 시스템 설계 사이의 관계에 관심을 둠
- 근원 추적성
  - 요구사항으로부터 요구사항을 제안한 stakeholder로의 링크
- 요구사항 추적성
  - 종속적인 요구사항 사이의 관계
- 설계 추적성
  - 요구사항에서 설계로의 링크


CASE 도구 지원

- 요구사항 저장소
  - 요구사항은 안전하게 관리되는 데이터 저장소에 보관됨.
- 변경 관리
  - 각 단계들 사이의 정보 흐름을 부분적으로 자동화
- 추적성 관리
  - 요구사항 사이의 링크들을 자동적으로 검색.


요구사항 변경 관리

- 제안된 모든 변경 사항을 요구사항에 적용하여야 함.
- 주요 단계
  - 문제 분석. 요구사항의 문제점을 토의하고 변경을 제안함.
  - 변경 분석과 비용 추정. 변경에 따른 다른 요구사항에의 영향을 측정.
  - 변경 실현. 요구사항 문서와 변경의 영향을 받는 다른 문서들을 수정.


요약

- 요구 공학 프로세스는 가능성 분석, 요구사항 유도와 분석, 요구사항 명세화, 그리고 요구사항 관리를 포함함.
- 요구사항 분석은 도메인 이해, 요구사항 수집, 분류, 구조화, 우선 순위화와 확인을 포함하는 반복 작업임.
- 시스템은 다른 요구사항을 가진 다수의 stakeholder들이 관련됨.

요약(계속)

- 사회적 조직적 요인들이 시스템 요구사항에
- 영향을 미침.
- 요구사항 검증은 정확성, 일관성, 완전성, 현실성, 그리고 시험 가능성에 대한 검사를 포함함.
- 기업 환경의 변화는 필연적으로 요구사항에 대한 변경을 가져옴.
- 요구사항 관리는 계획과 변경 관리를 포함함.