



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.



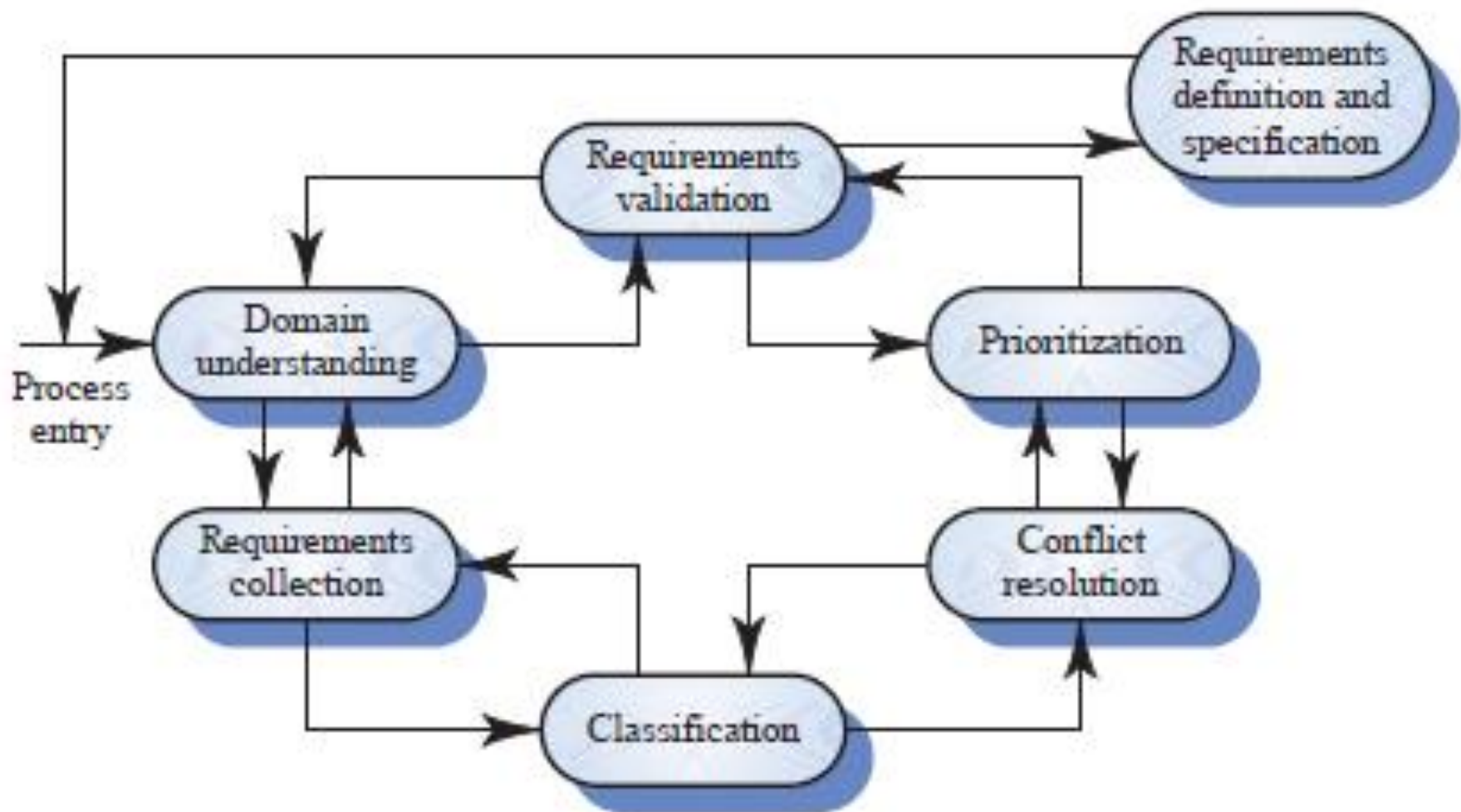


요구사항 분석의 문제점

- ❖ 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





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.



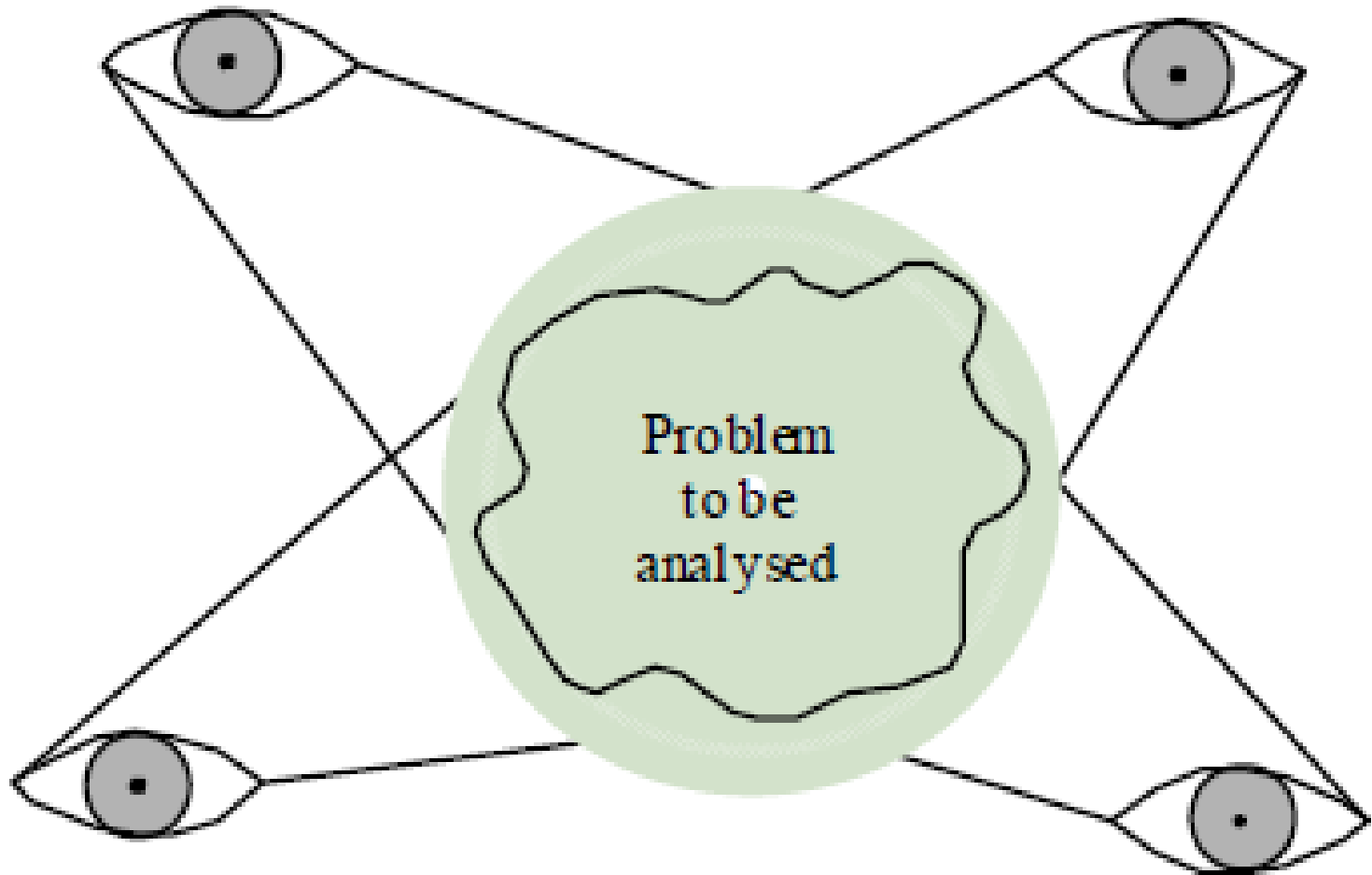


관점 중심의 유도

- ❖ 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 입장에서 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)





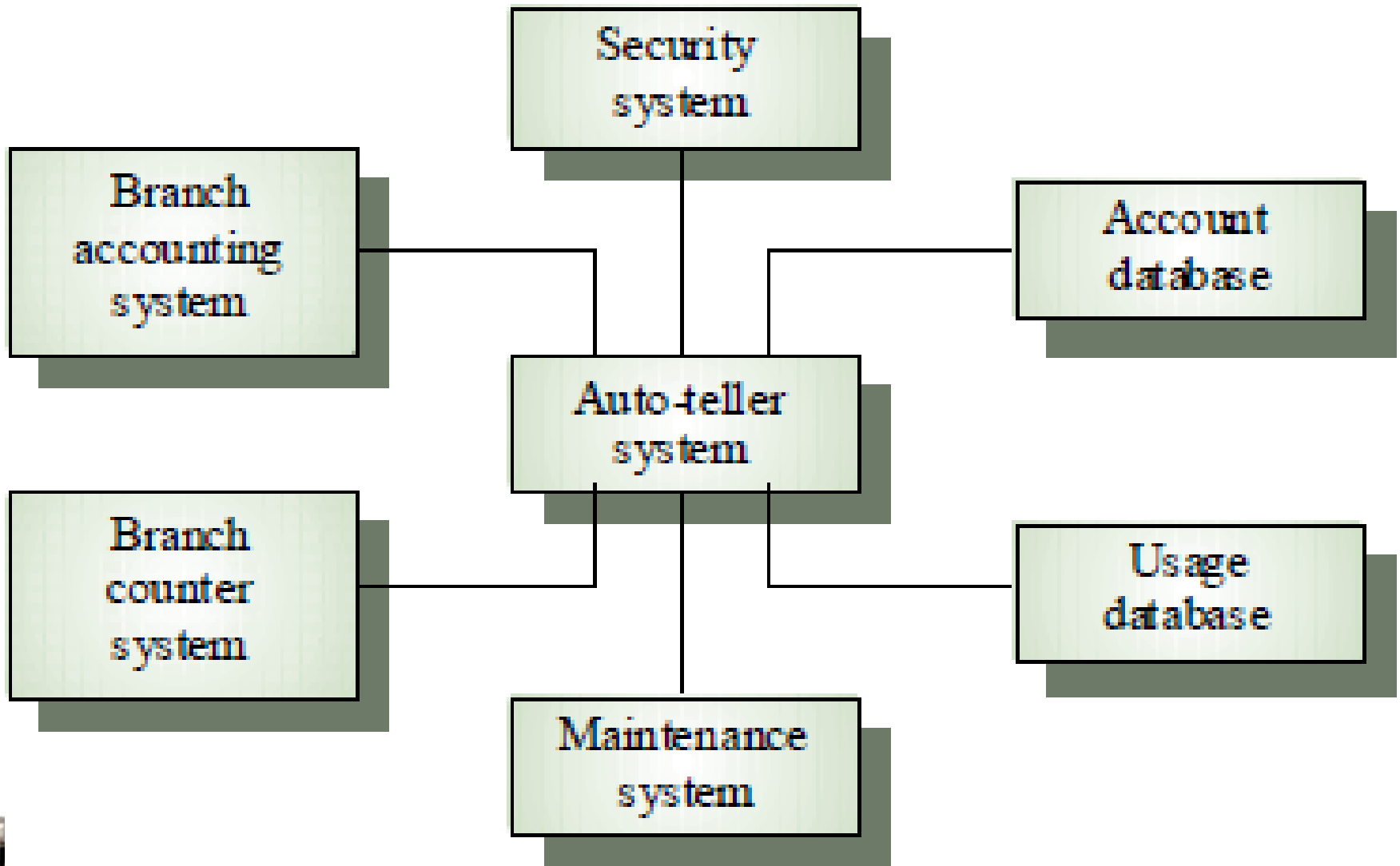
System 범위(contexts)

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





Auto-teller system context



Scenarios



- ❖ 시나리오에는 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.
- ❖ 시나리오에는 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



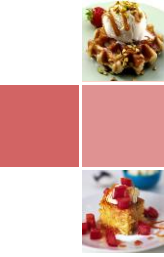


이벤트 시나리오

- ❖ '트랜잭션 시작'과 같은 어떤 특정한 이벤트 발생에 대해 시스템이 어떻게 반응하는가를 설명하는데 사용됨.
- ❖ 이벤트 시나리오에 사용되는 다이어그램 규칙 (VORD 의 경우)
 - 제공되고 전달되는 데이터
 - 제어정보
 - 예외 처리
 - 예상되는 다음이벤트



데이터와 제어 분석을 위한 표기법



- ❖ 타원. 관점으로로부터 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.



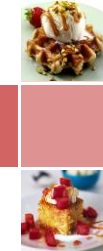
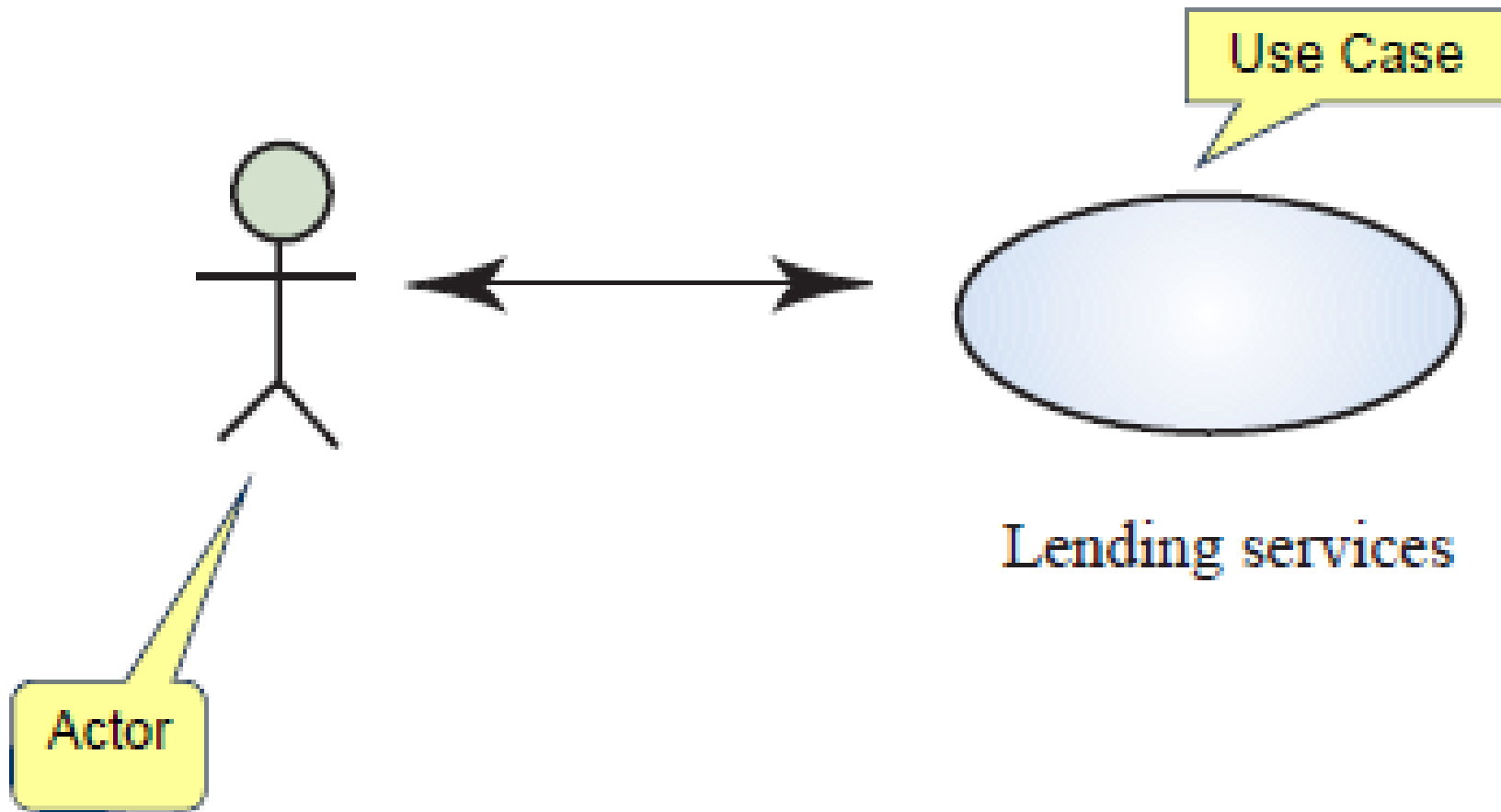


Use cases

- ❖ **Use-cases**는 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 다이어그램(**Sequence diagrams**)이 **use case**에 상세한 설명을 추가하기 위해 사용될 수 있음.

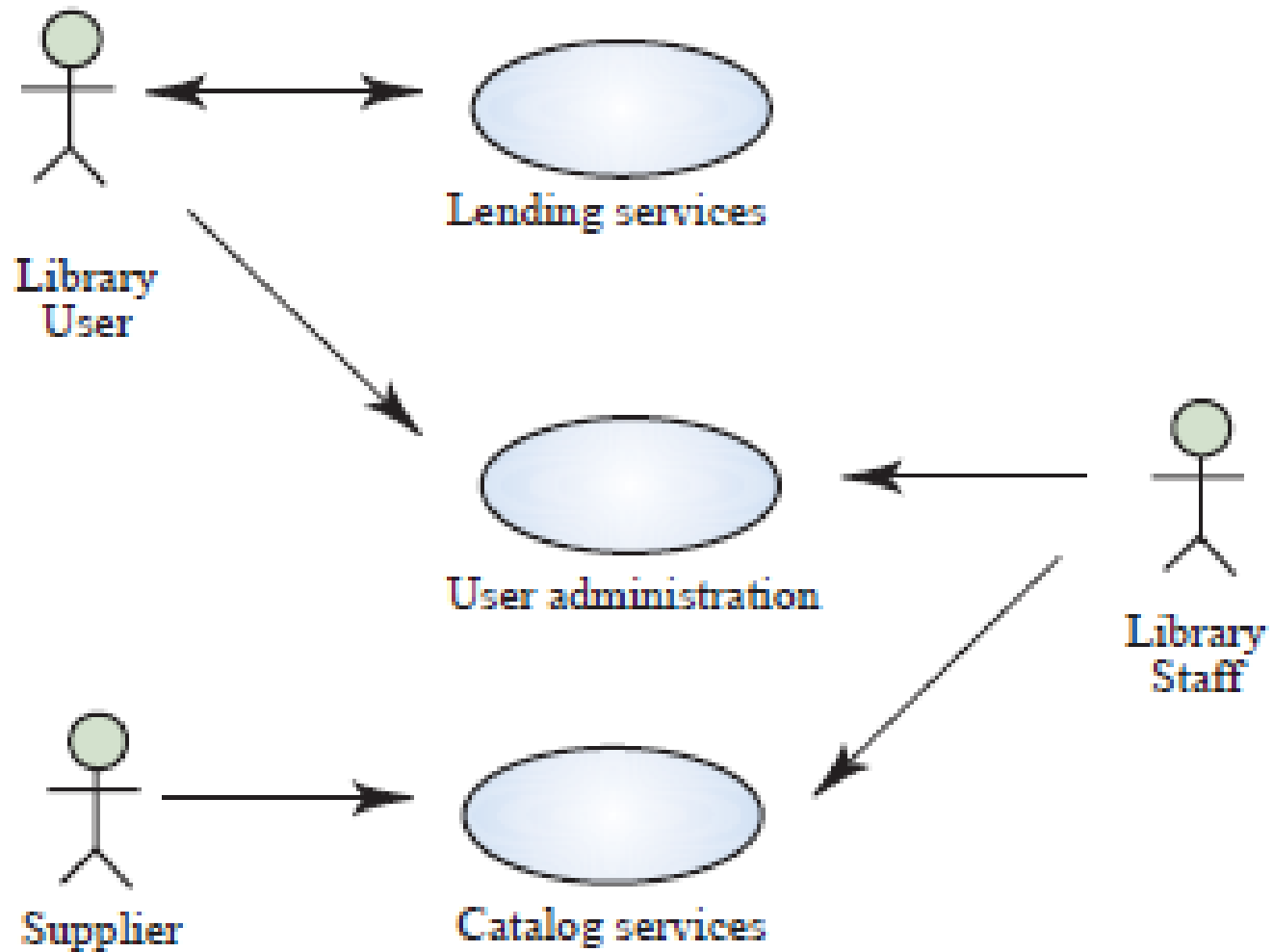


대출(Lending) use-case





도서관 시스템에 대한 usecases



Catalogue management-Sequence diagram

