



System models

- ❖ 요구사항을 분석중인 시스템에 대한 추상화된 표현





Objectives

- ❖ 시스템의 범위(context)가 RE 프로세스의 일부 분으로 모델링되어야 하는 이유를 설명.
- ❖ behavioural modelling, data modelling and
- ❖ object modelling을 기술.
- ❖ Unified Modeling Language (UML)에서 사용되는 일부 표기법에 대한 소개
- ❖ CASE workbenches들이 system modelling을 지원하는 방법을 보여줌.





System modeling

- ❖ System modeling은 분석가가 시스템의 기능을 이해하는 것을 도와주며 모델들은 고객들과 정보를 교환하기 위해 사용됨.
- ❖ 여러 모델들이 다른 관점으로 시스템을 표현함.
 - 시스템의 범위 또는 환경을 보여주는 external perspective
 - 시스템의 행동(동작)을 보여주는 Behavioral perspective
 - 시스템 구조 또는 데이터 구조를 보여주는 Structural perspective



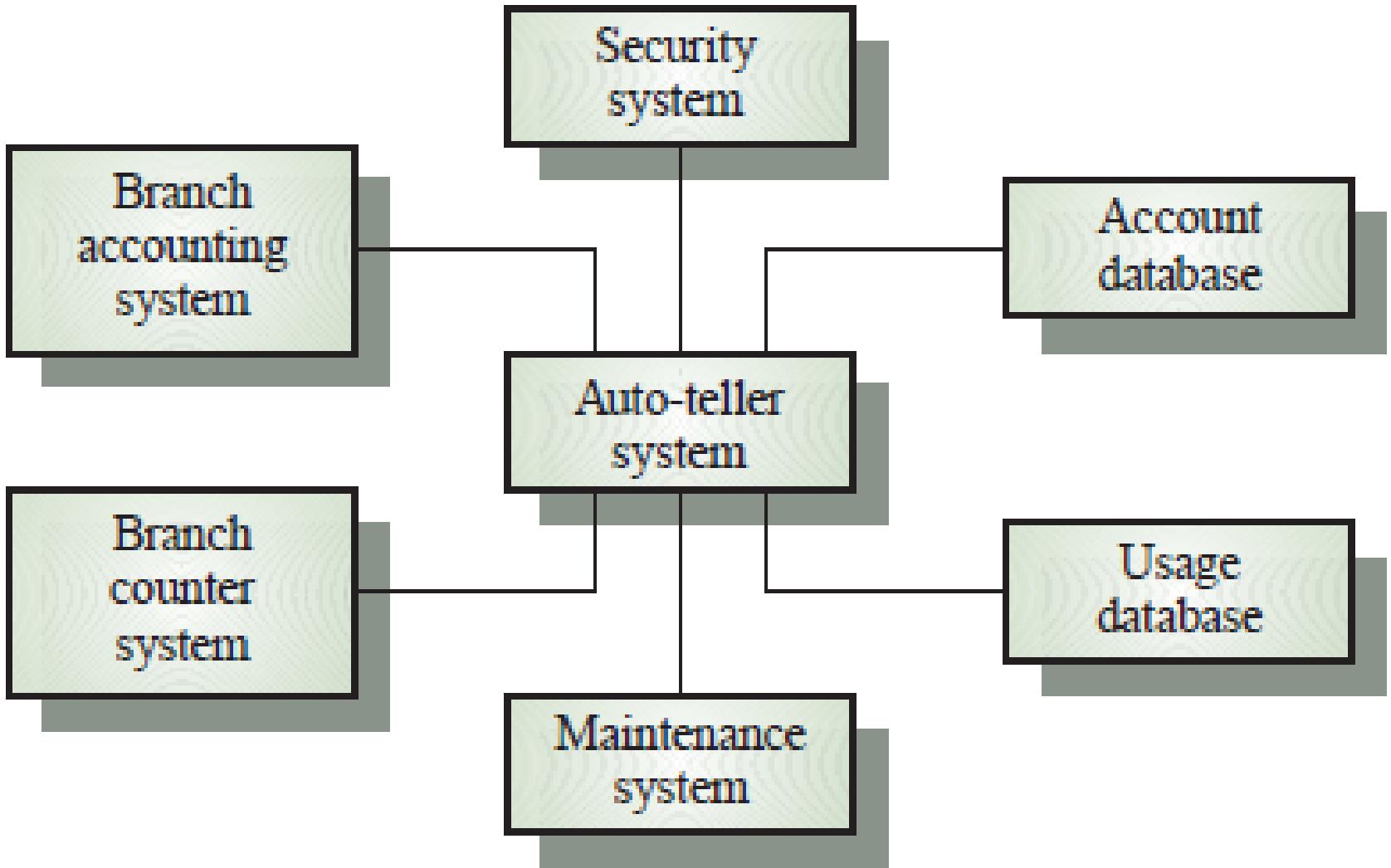


Context models

- ❖ Context model들은 시스템의 범위를 설명하기 위해 사용됨.
- ❖ 사회와 조직의 관심이 시스템의 범위를 결정하는 데 영향을 줄 수 있음.
- ❖ Architectural model들은 시스템, 그리고 다른 시스템과의 관계를 보여줌.



The context of an ATM system



Process models

- ❖ Process model들은 전반적인 프로세스, 그리고 시스템이 지원하는 프로세스들을 보여줌.
- ❖ 자료 흐름(Data flow) model은 프로세스들, 그리고 한 프로세스에서 다른 프로세스로의 정보의 흐름을 보여주는데 사용될 수 있음.





Behavioural models

- ❖ 시스템의 전반적인 행동을 묘사하기 위해 사용됨.
- ❖ □ Two types of behavioural model are shown here
 - 데이터가 시스템에서 어떻게 이동하며 처리되는지를 보여주는 Data processing model
 - 이벤트에 대한 시스템의 반응을 보여주는 State machine model
- ❖ 두개의 모델 다 시스템의 행동에 대한 설명을 위해 필요함.

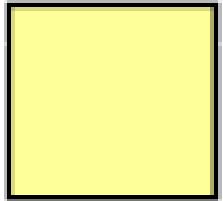


Data-processing models

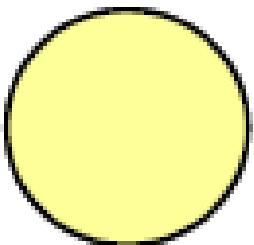
- ❖ 자료 흐름도(Data flow diagram)가 시스템의 데이터 처리 과정을 모델링하기 위해 사용됨.
- ❖ 이것은 데이터가 시스템에서 흘러가면서 처리되는 단계를 보여줌.
- ❖ 많은 분석 모델에서 기본적으로 포함됨.
- ❖ 고객이 이해할 수 있는 단순하고 직관적인 표기법임.
- ❖ 데이터의 end-to-end processing을 보여줌.



Flow Modeling Notation



external entity



process



data flow



data store



External Entity

- ❖ A producer or consumer of data
- ❖ Examples: a person, a device, a sensor
- ❖ Another example: computer-based system
- ❖ *Data must always originate somewhere*
- ❖ *and must always be sent to something*



Process

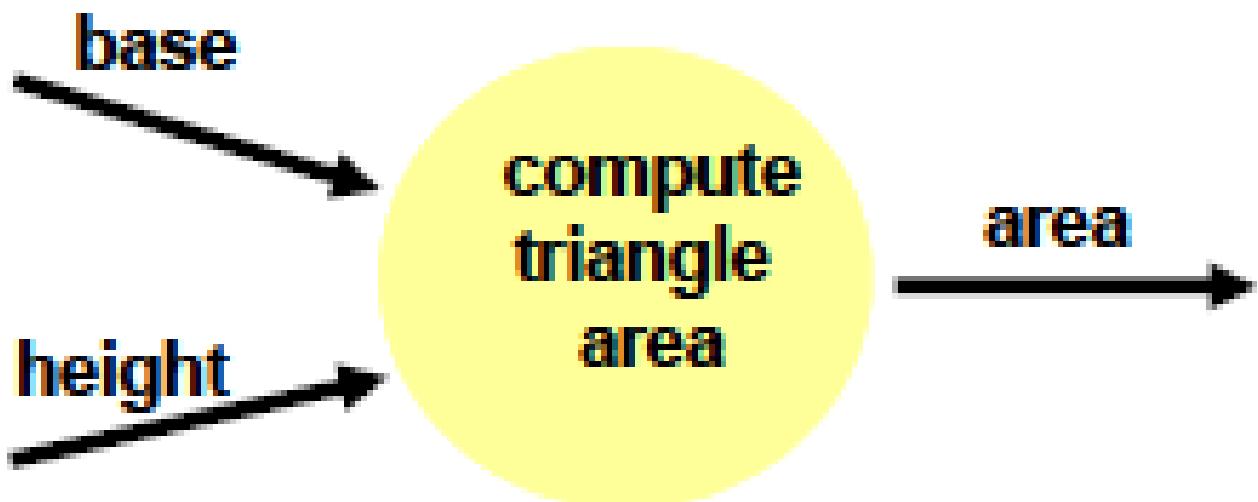
- ❖ A data transformer (changes input to output)
- ❖ Examples: compute taxes, determine area, format report, display graph
- ❖ *Data must always be processed in some way to achieve system function*





Data Flow

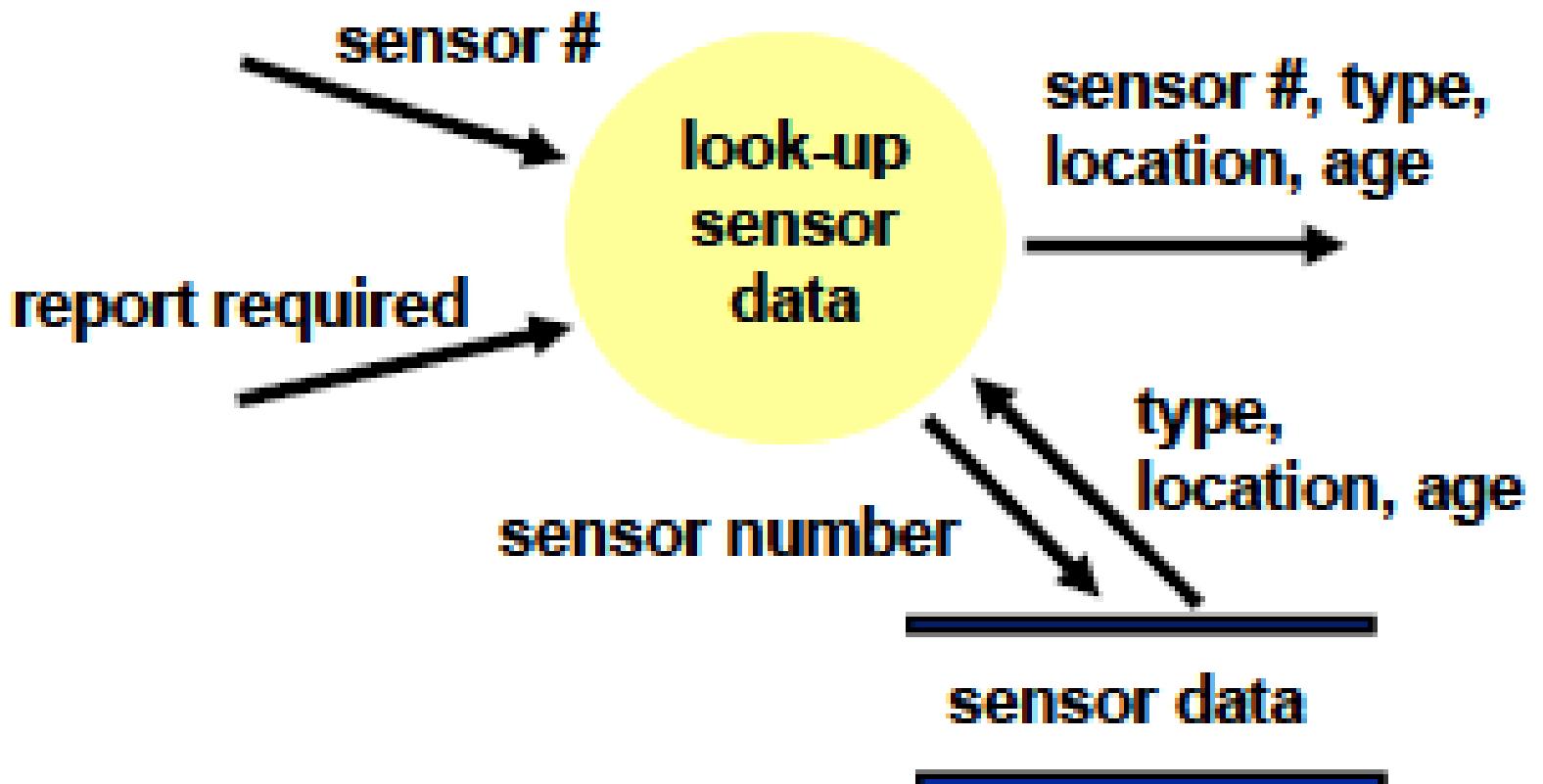
- ❖ Data flows through a system, beginning as input and being transformed into output.





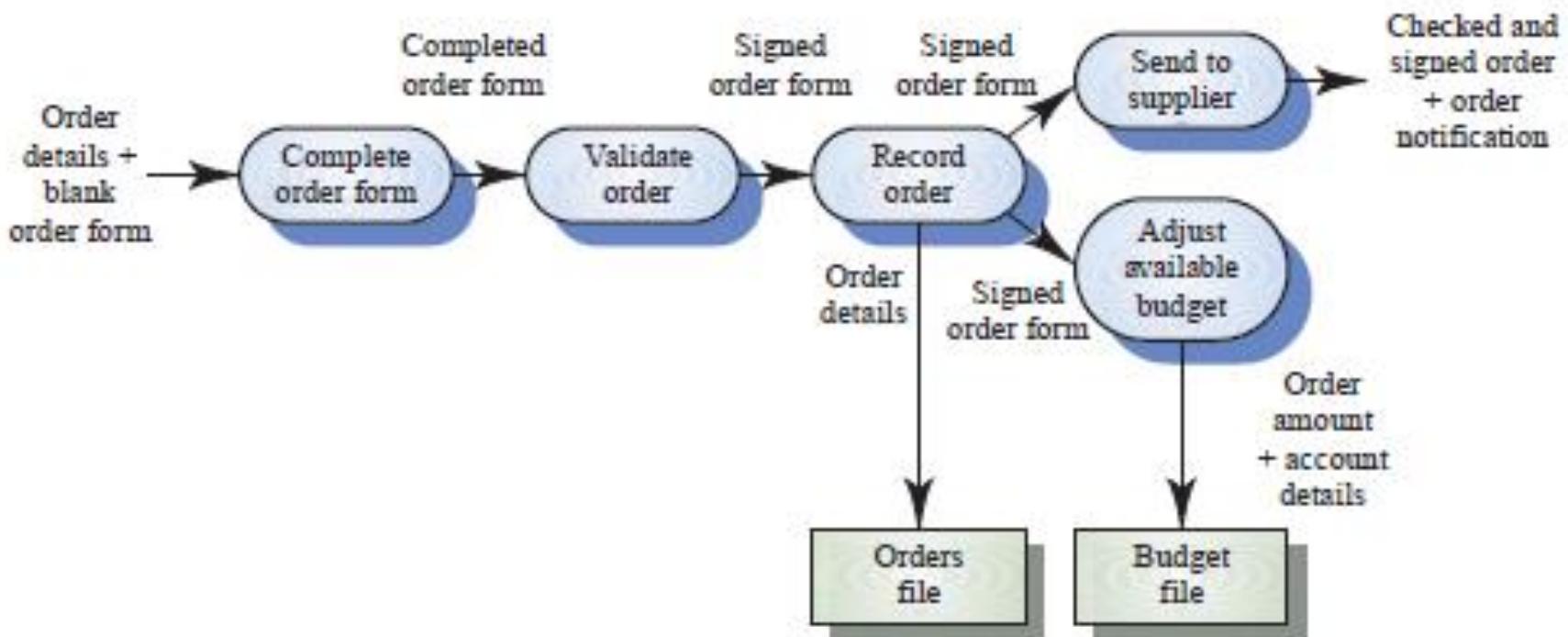
Data Stores

- ❖ Data is often stored for later use.





Order processing DFD





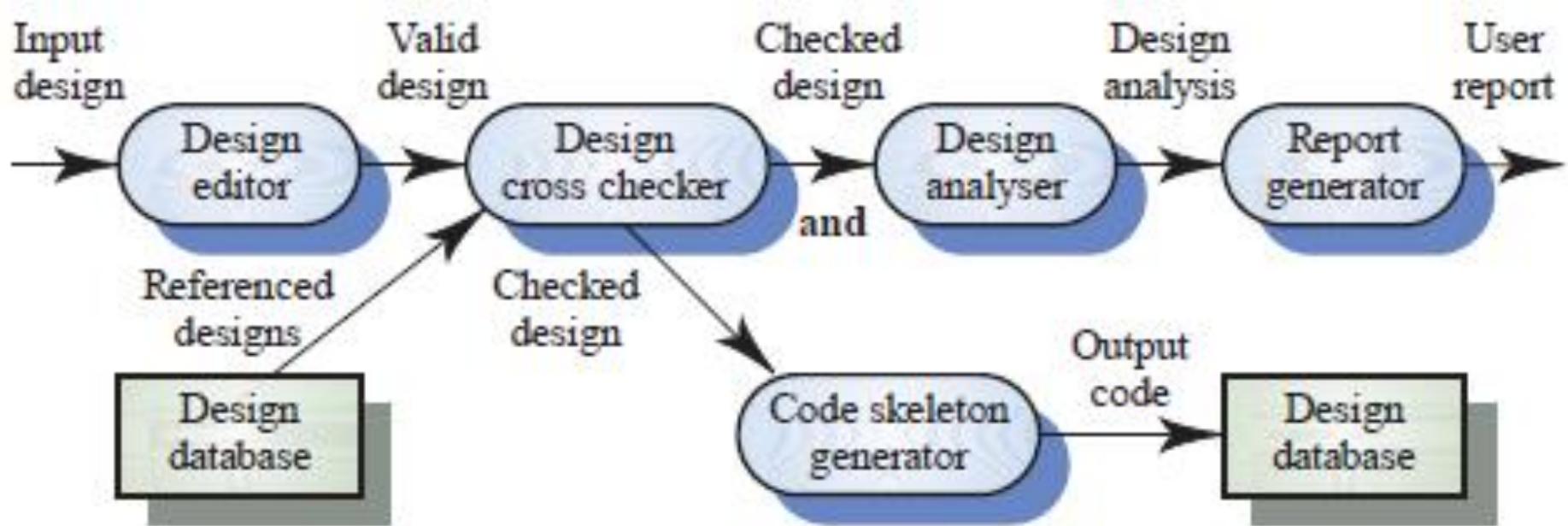
Data flow diagrams

- ❖ 기능적 관점에서 시스템을 모델링 함.
- ❖ 데이터가 프로세스에 어떻게 관련되어 있는지를 추적하고 문서화하는 것은 시스템에 대한 전반적인 이해에 도움을 줌.
- ❖ 자료 흐름도는 시스템과 환경내의 다른 시스템 사이의 데이터 교환을 보여주는 데에도 사용될 수 있음.



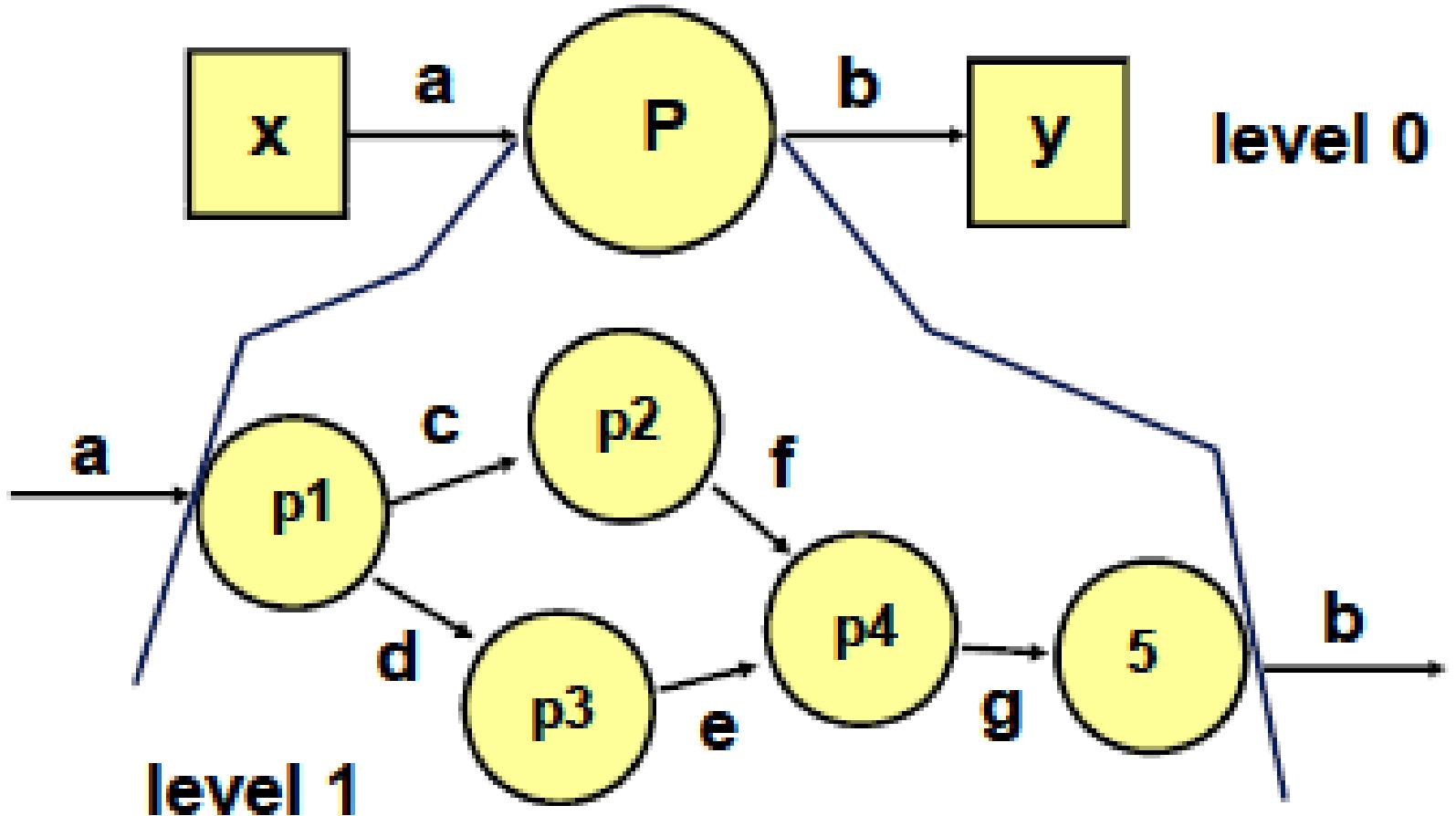


CASE toolset DFD





The Data Flow Hierarchy



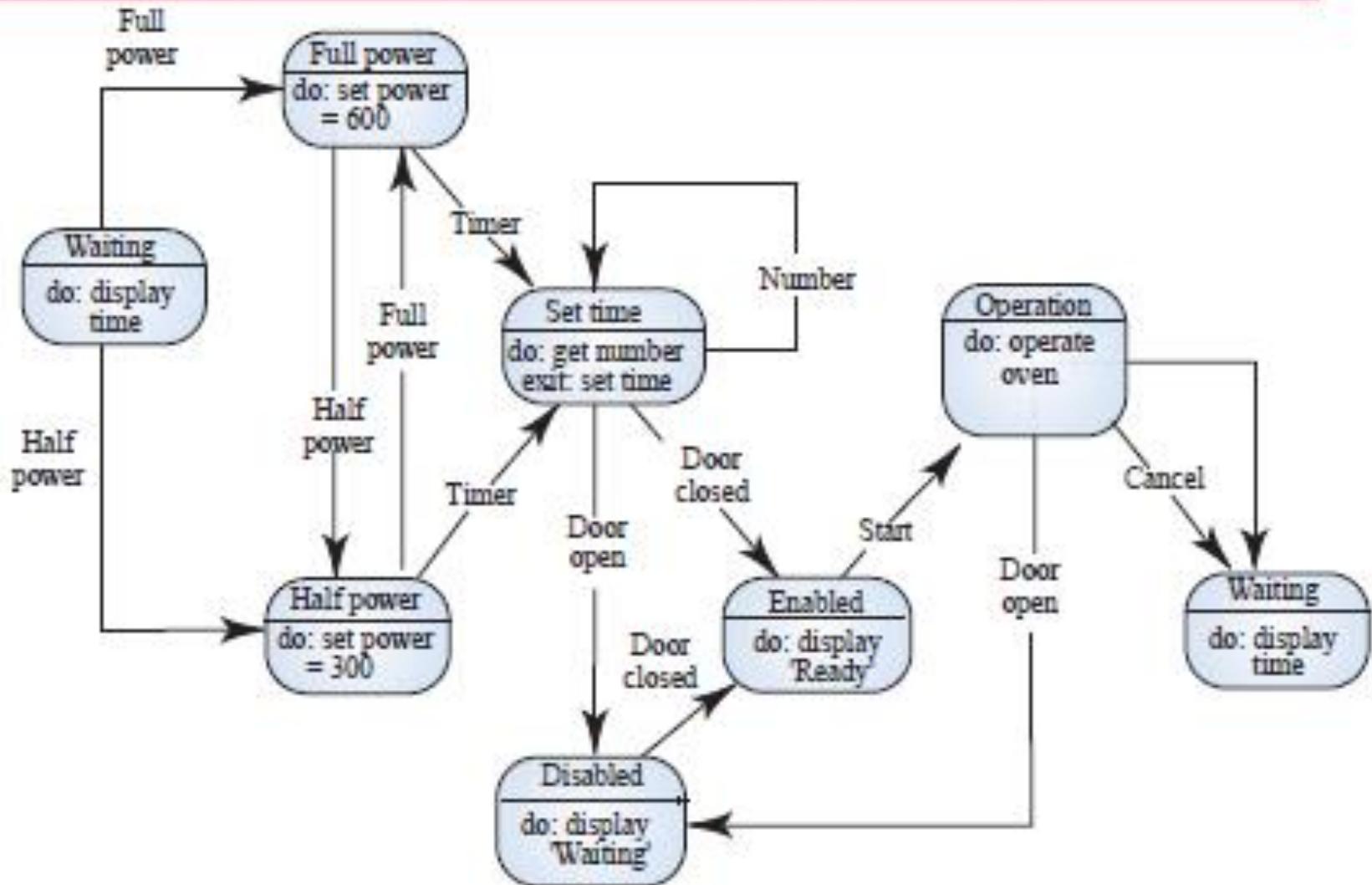
State machine models

- ❖ 이것은 외부 또는 내부 이벤트에 대한 반응으로 수행하는 시스템의 행동을 모델링 함.
- ❖ 자극에 대한 시스템의 반응을 보여주므로 종종 실시간 시스템을 모델링하기 위해 사용됨.
- ❖ 이 모델은 시스템의 상태를 노드로 그리고 이벤트를 노드 사이의 간선으로 모델링 함. 이벤트가 발생하면, 시스템은 어떤 상태에서 다른 상태로 이동함.
- ❖ Statechart는 UML의 핵심적인 부분임.





Microwave oven model



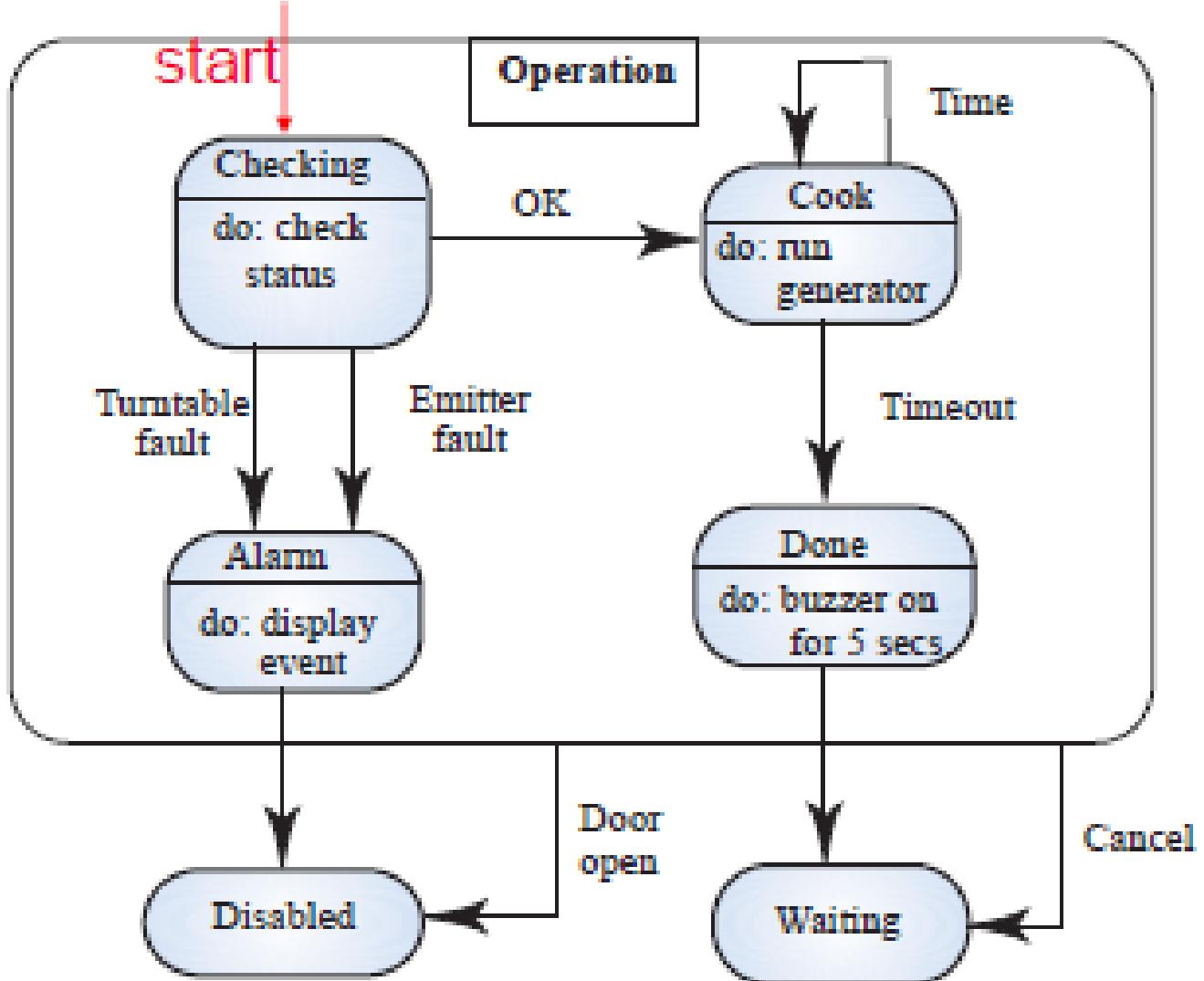


Statecharts

- ❖ 모델을 서브 모델들로 분해하는 것을 허용 함.(see following slide)
- ❖ 동작(action)에 대한 간단한 설명이 각 상태에서 'do' 다음에 포함됨.
- ❖ 상태와 자극(stimuli)을 설명하는 표로 보충될 수 있음.



Microwave oven operation



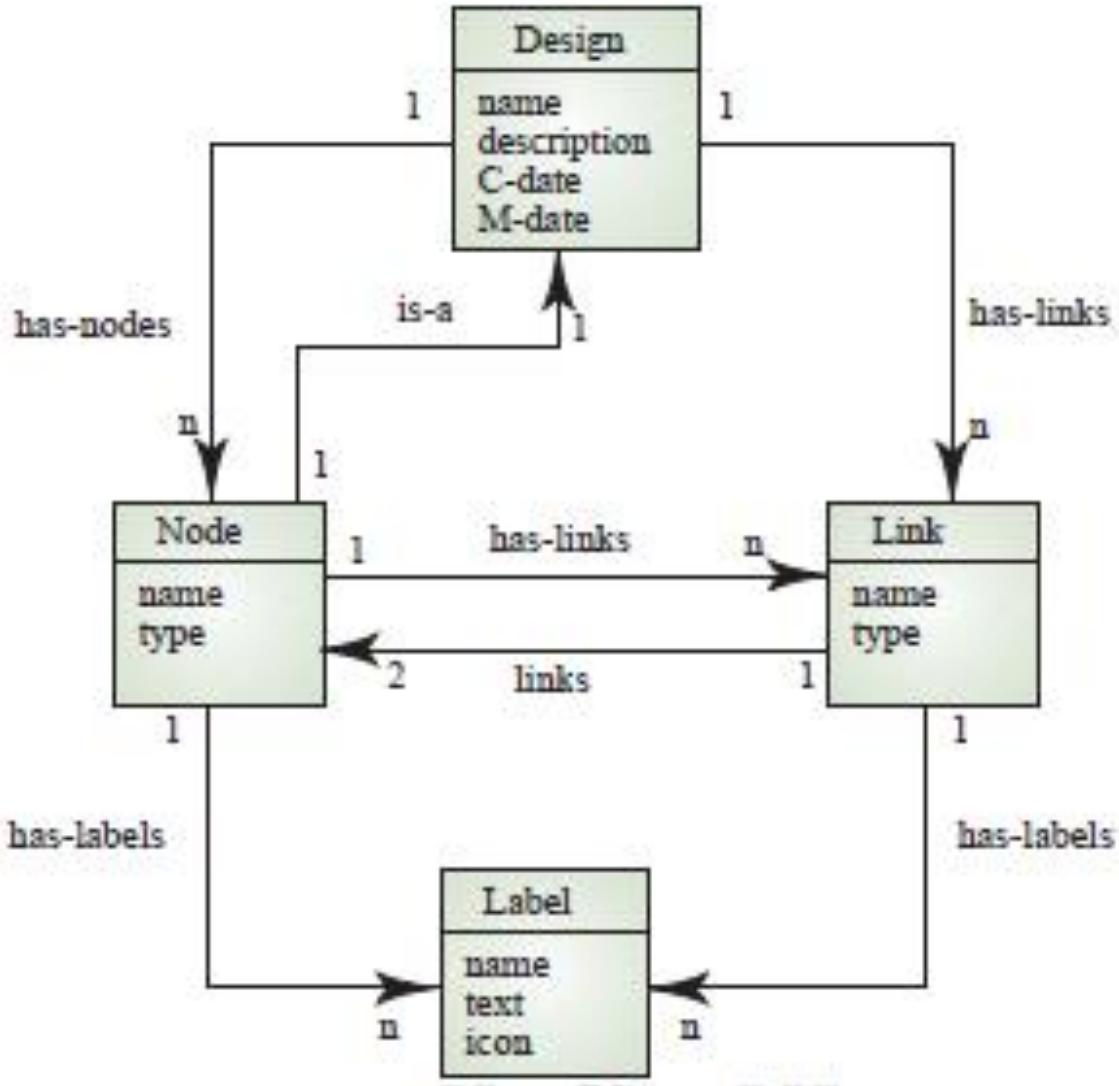


Semantic data models

- ❖ 시스템에 의해 처리되는 데이터의 logical 구조를 묘사하기 위해 사용됨.
- ❖ Entity-relation-attribute 모델은 시스템에 있는 개체(entity), 개체들 사이의 관계, 그리고 개체의 속성들을 설명함.
- ❖ 데이터베이스 설계에서 널리 사용됨. 관계형 데이터베이스를 사용하여 쉽게 구현될 수 있음.
- ❖ UML에서는 특정한 표기법을 제시하지 않음. 그러나 객체들과 접속(associations)은 사용될 수 있음.



SW 설계에 대한 semantic model





Data dictionaries

❖ 자료 사전(Data dictionary)은 시스템 모델들에서 사용되는 모든 이름들에 대한 목록임. 개체, 관계, 그리고 속성들에 대한 설명도 포함됨.

❖ Advantages

- 이름에 대한 관리를 지원하여, 중복(duplication)을 피할 수 있음.
- 분석, 설계, 그리고 구현을 연결하는 조직의 지식에 대한 저장소

❖ 많은 CASE workbench들이 자료사전 기능을 지원함.





Object models

- ❖ 객체 모델은 시스템을 객체 클래스의 용어로서 기술함. **Class = attribute + method**
- ❖ 객체 클래스는 각각의 객체가 제공하는 공통의 속성들과 서비스들(동작들)을 가진 객체들의 집합에 대한 추상화임.
- ❖ 다양한 객체 모델들이 생성될 수 있음.
 - Inheritance models 상속: is-a
 - Aggregation models 집단화: part-of
 - Interaction models





Object models (계속)

- ❖ 시스템에 의해 다루어지는 실세계의 개체들을 반영하는 자연스러운 방법임.
- ❖ 이 방법을 사용하여 추상적인 개체들을 모델링하는 것은 조금 어려움.
- ❖ 객체 클래스 인식은 어플리케이션 도메인에 대한 깊은 지식을 필요로 하는 어려운 프로세스로 인식됨 - 문법적분석, 역할...
- ❖ 도메인 개체들을 반영하는 객체 클래스들은 여러 시스템에서 재사용될 수 있음,



Inheritance models

- ❖ 객체 클래스들을 계층적으로 구성함.
- ❖ 계층의 최상위에 있는 클래스들은 모든 클래스들의 공통적인 특성을 반영함.
- ❖ 객체 클래스들은 하나 이상의 **super generalization**
- ❖ 클래스들로부터 속성과 서비스들을 상속 받음.
필요하다면 이것들을 특징화(specialization)할 수 있음.
- ❖ 여러 브랜치에서의 중복을 피하도록 클래스 계층을 설계하는 것은 어려움.





The Unified Modeling Language

- ❖ 널리 사용되던 객체 지향 분석과 설계 방법의 개발자들이 제안함.
- ❖ 객체 지향 모델링에 대한 사실상의 표준이 되었음.

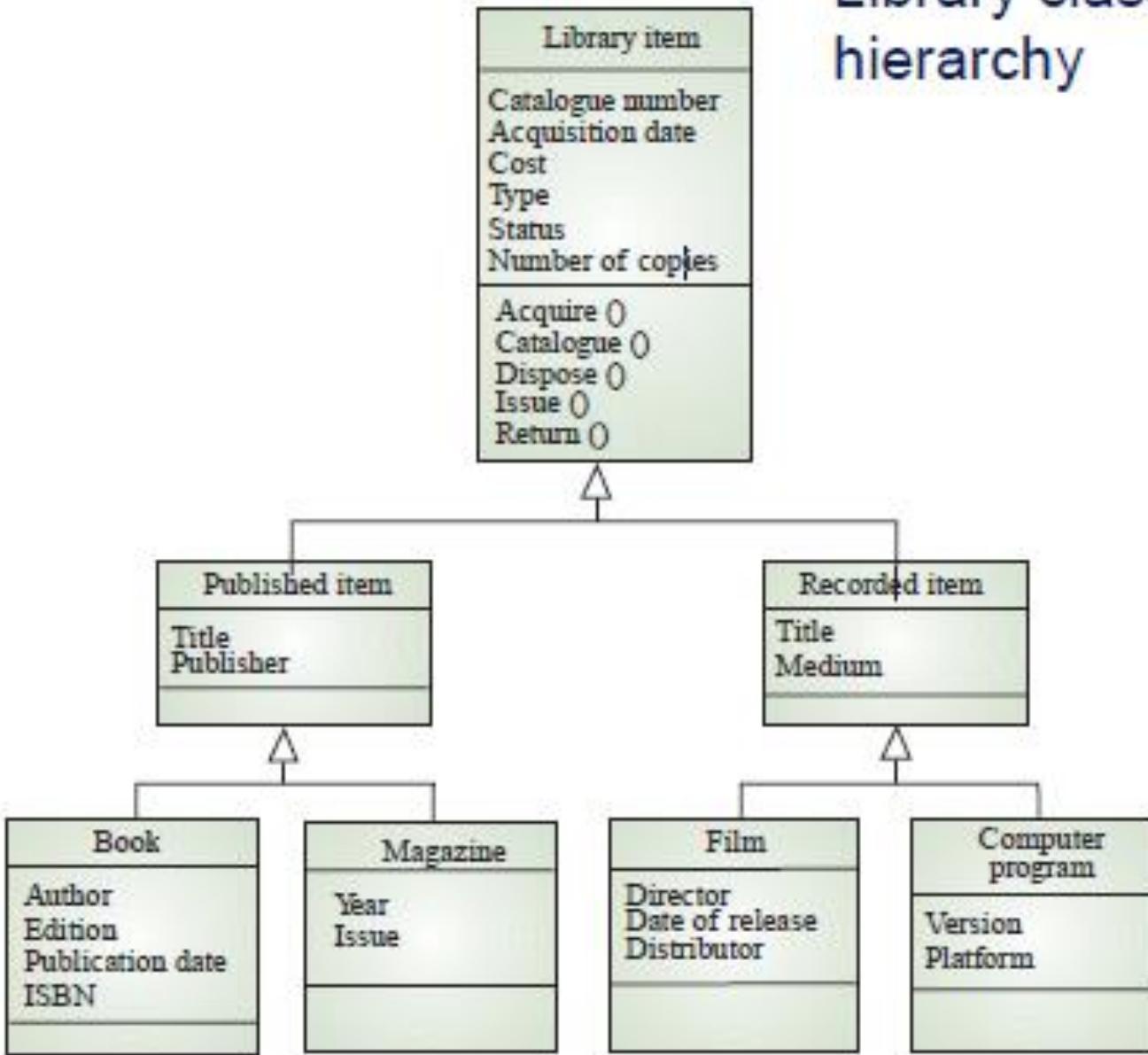
❖ Notation

- 객체 클래스는 사각형(위 부분에는 이름, 중간에는 속성, 아래 부분에는 동작들)으로 표현.
- 객체 클래스들 사이의 관계(접속)는 객체들을 연결하는 라인들로 나타냄.
- 상속성은 일반화(generalisation)를 나타내며, 계층에서 위로 향하는 화살표 모양으로 표기됨.



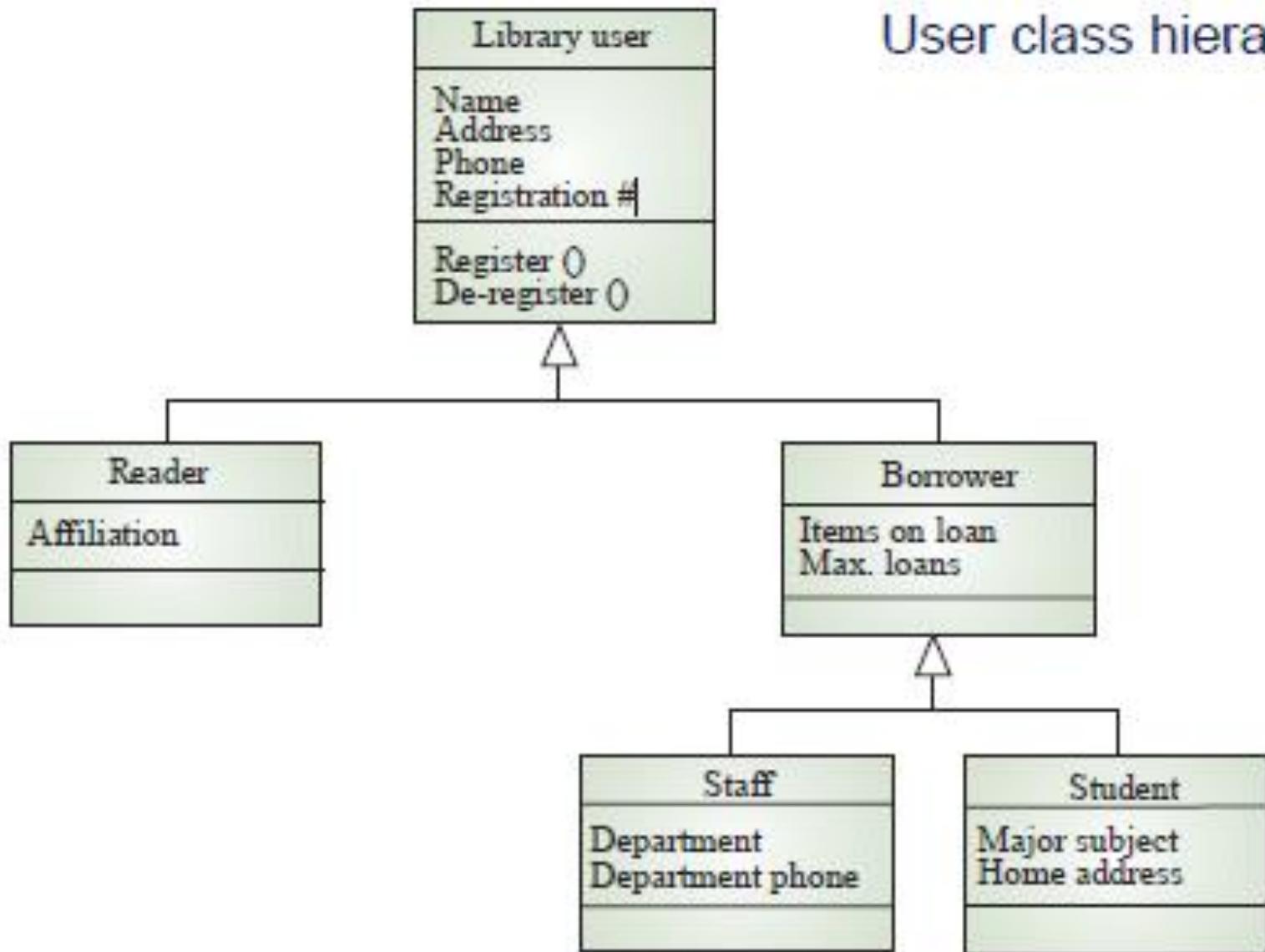


Library class hierarchy





User class hierarchy



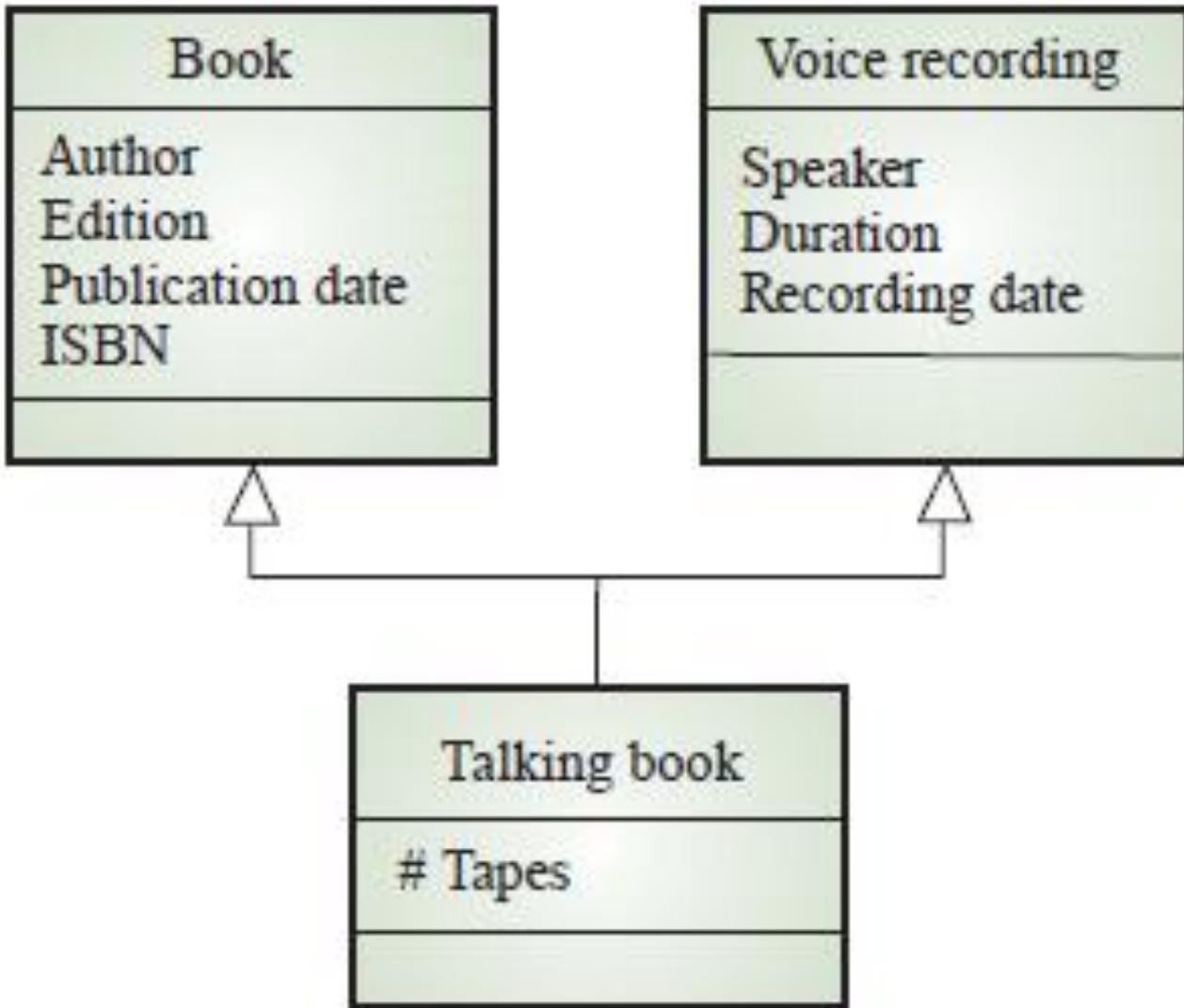


Multiple inheritance

- ❖ 속성들과 서비스들을 하나의 `super` 클래스가 아닌 여러 `super` 클래스들로부터 상속 받음.
- ❖ 여러 `super` 클래스에서 동일한 이름의 속성/서비스들이 다른 semantic을 갖는다면, `semantic conflicts`가 발생할 수 있음.
- ❖ 클래스 계층의 재구성이 더욱 복잡해짐.



Multiple inheritance





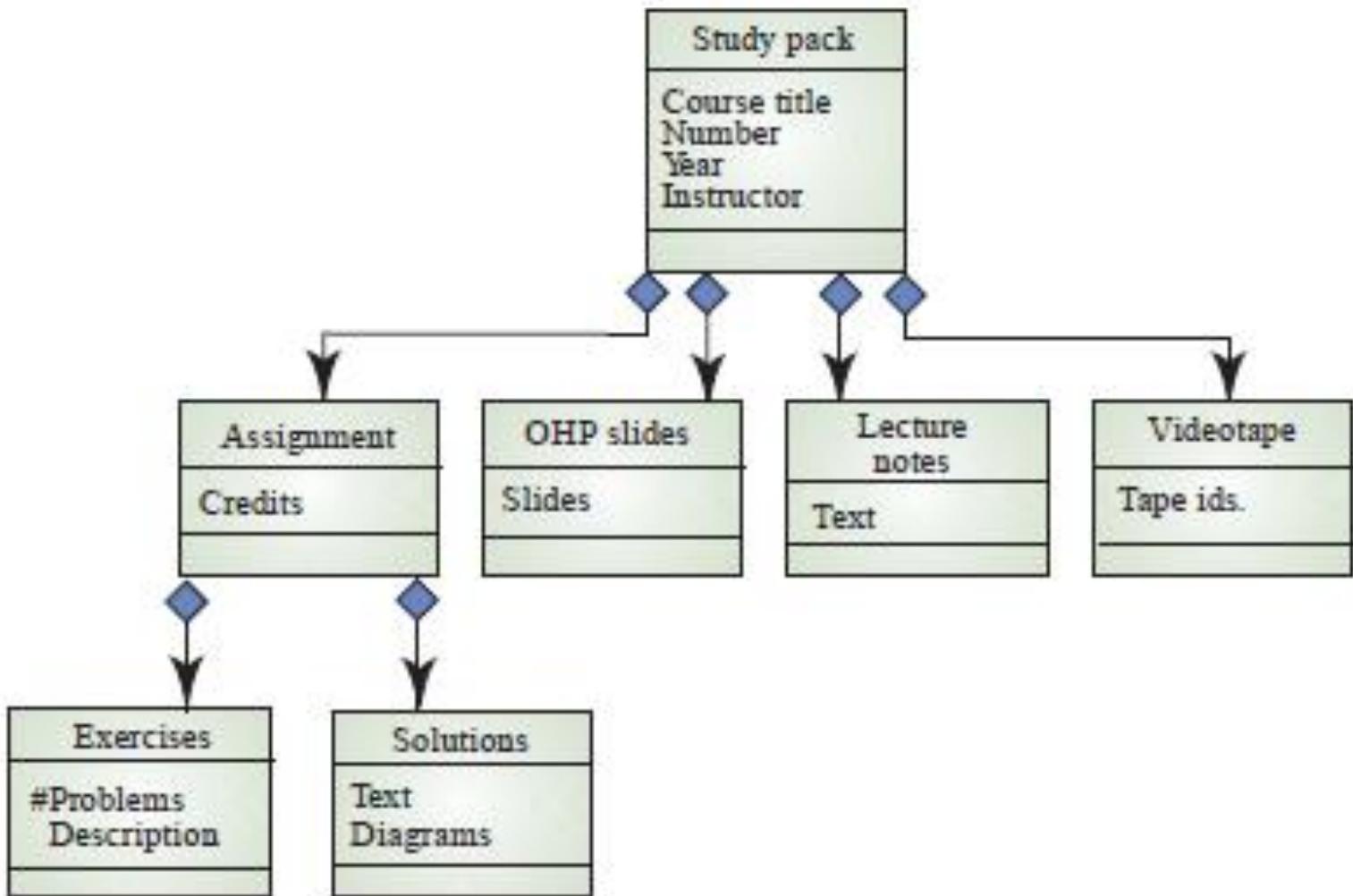
Object aggregation

- ❖ **Aggregation model**은 클래스 모음인 클래스가 어떻게 다른 클래스들로 구성되어 있는지를 보여줌.
- ❖ **semantic data model**의 **part-of** 관계와 비슷함.





Object aggregation



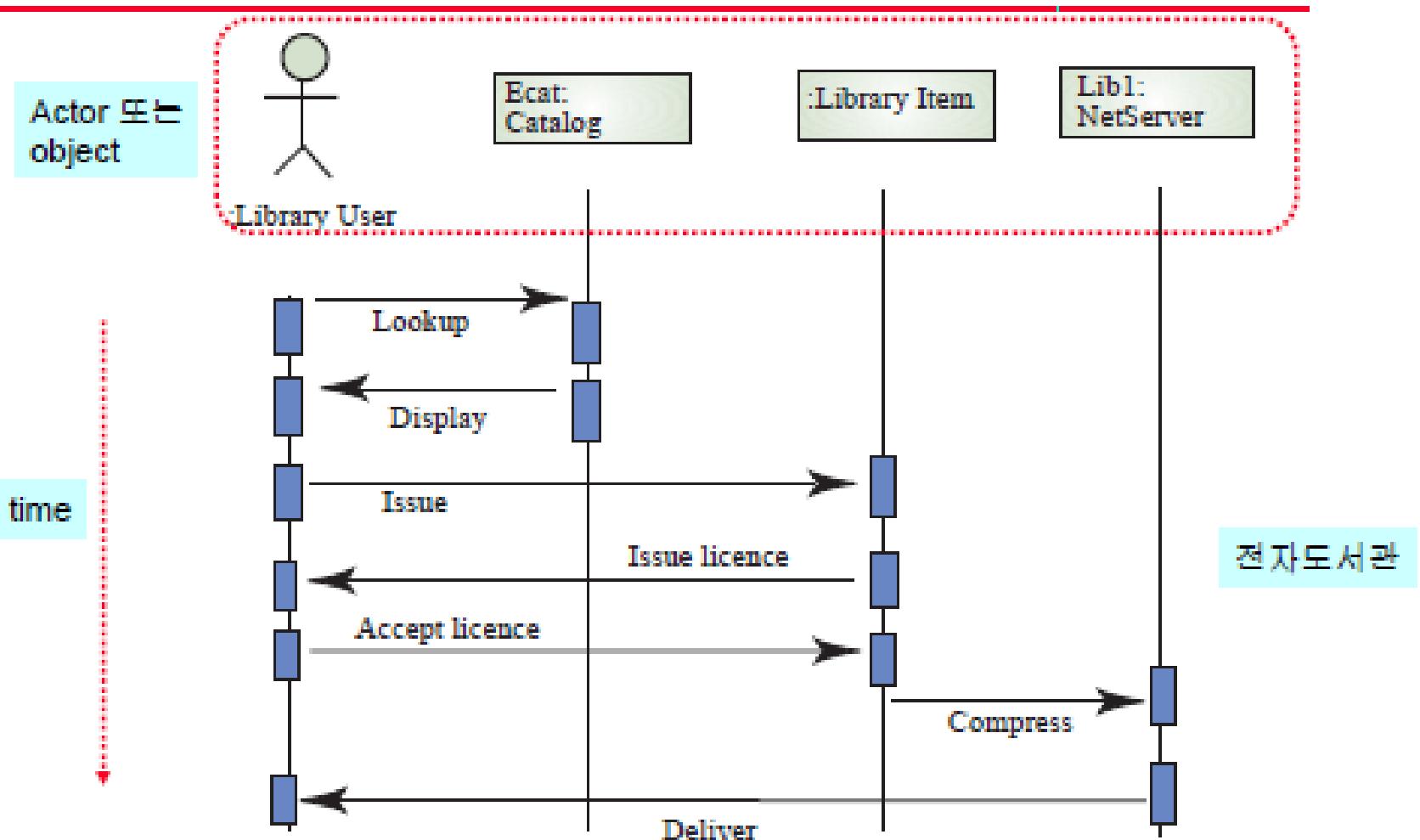


Object behaviour modelling

- ❖ Behavioural model은 use-case로 명시된 특정한 시스템의 행동을 일으키는 객체들 사이의 상호작용을 보여줌-semantic equivalent
- ❖ UML의 Sequence diagrams (또는 collaboration diagrams)이 객체들 사이의 상호작용을 모델링 하는데 사용됨.



Issue of electronic items



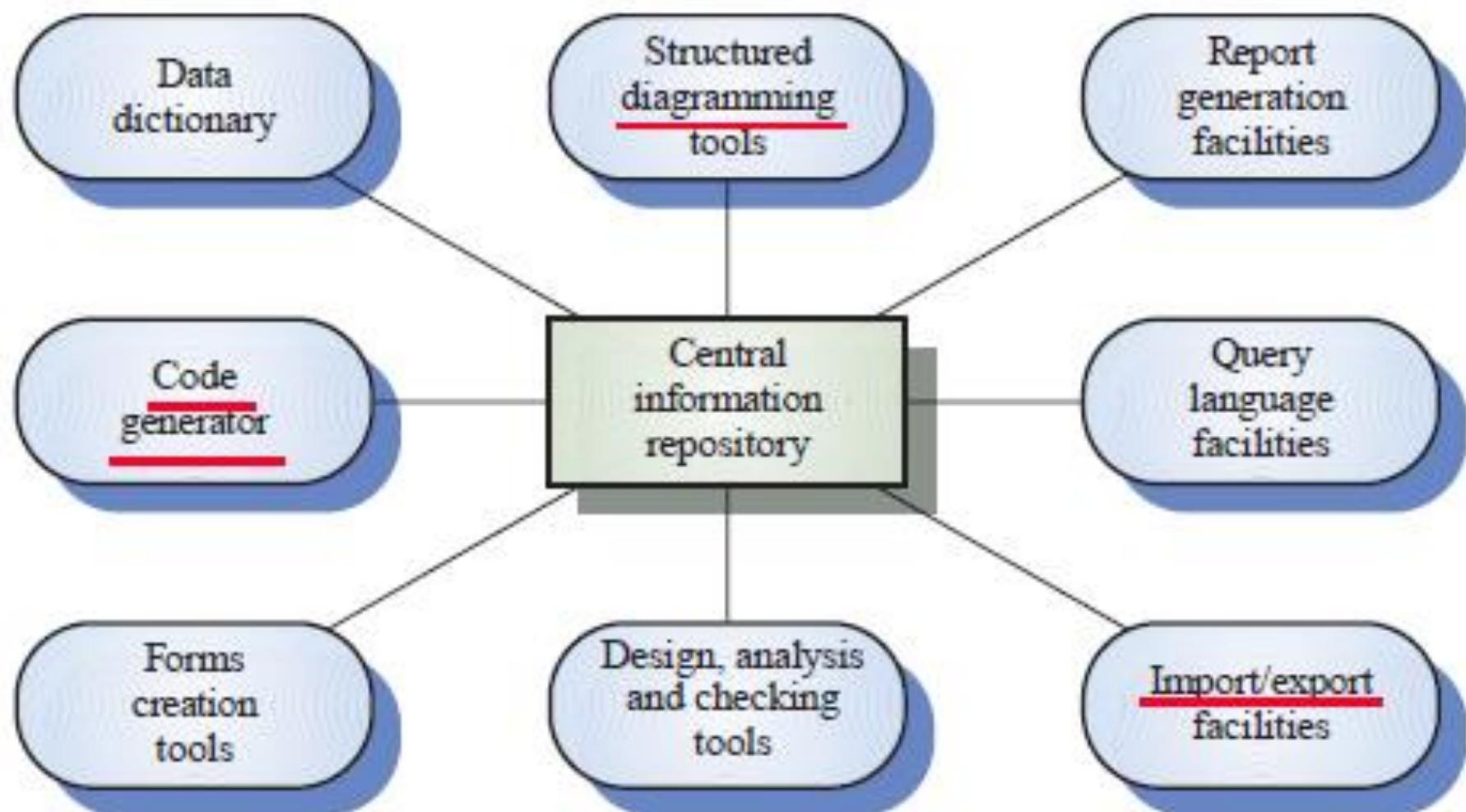
CASE workbenches

- ❖ 분석, 설계 또는 테스팅과 같이 관련되어 있는 소프트웨어 프로세스 활동들을 지원하기 위해 고안된 도구들의 집합.
- ❖ 분석과 설계 워크벤치들은 요구 공학과 시스템 설계 단계 동안 시스템 모델링을 지원함.
- ❖ 이런 워크벤치들은 특정한 설계 방법을 지원하거나 또는 여러 유형의 시스템 모델들을 생성하는 것을 지원함.





An analysis and design workbench



Analysis workbench components

- ❖ Diagram editors
- ❖ 모델 분석과 검사 도구
- ❖ 정보저장소(Repository)와 query language
- ❖ Data dictionary
- ❖ 보고서 정의와 생성 도구
- ❖ Forms definition tools
- ❖ Import/export translators
- ❖ Code generation tools





Key points

- ❖ 모델은 시스템에 대한 추상적 관점임. 여러 유형의 모델이 다른 시스템 정보를 제공함(보완적으로).
- ❖ **Context model**은 환경에서 시스템의 위치를 다른 시스템들, 프로세스들과 함께 보여줌.
- ❖ **Data flow model**은 시스템에서의 자료 처리를 모델링 하는데 사용될 수 있음.
- ❖ **State machine model**은 내부 또는 외부 이벤트에 대한 응답으로 수행하는 시스템의 동작을 모델링 함.





Key points

- ❖ Semantic data model은 시스템에서 사용 (import 또는 export)하는 데이터의 논리적 구조를 묘사함.
- ❖ Object model은 시스템의 논리적 개체들, 그것들에 대한 분류와 집단화(aggregation)를 설명 함.
- ❖ CASE 워크벤치는 시스템 모델의 개발을 지원함.

