Chapter 7

System Models

Objectives

- To explain why the context of a system should be modelled as part of the RE process
- To describe behavioural modelling, data modelling and object modelling
- To introduce some of the notations used in the Unified Modeling Language (UML)
- To show how CASE workbenches support system modelling

Topics covered

- Context models
- Behavioural models
- Data models
- Object models

System modelling

- System modelling helps the analyst to understand the functionality of the system and models are used to communicate with customers
- Different models present the system from different perspectives
  - External perspective showing the system's context or environment
  - Behavioural perspective showing the behaviour of the system
  - Structural perspective showing the system or data architecture

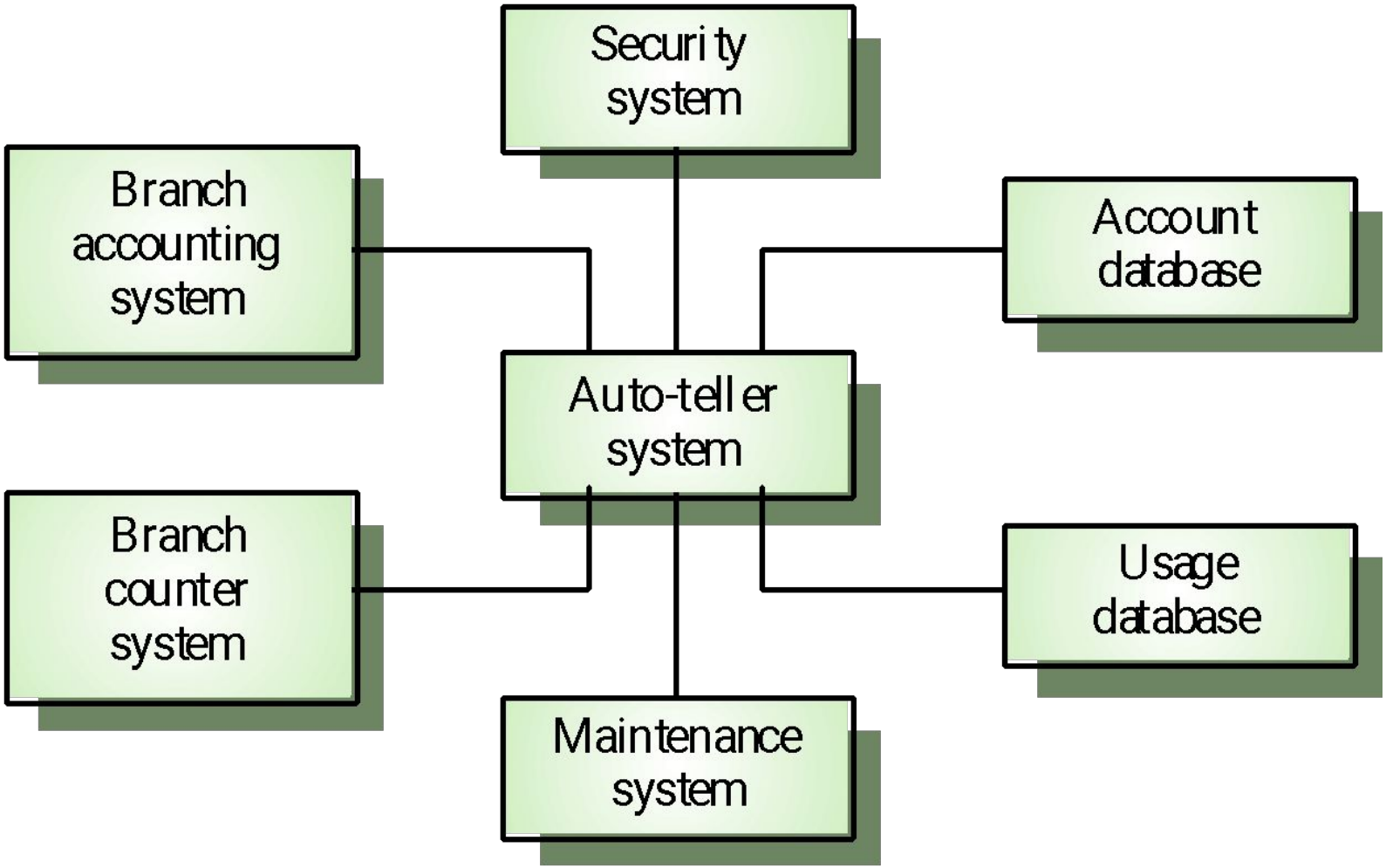
Model types

- Data processing model showing how the data is processed at different stages
- Composition model showing how entities are composed of other entities
- Architectural model showing principal sub-systems that make up the system.
- Classification model showing how entities have common characteristics
- Stimulus/response model showing the system's reaction to events

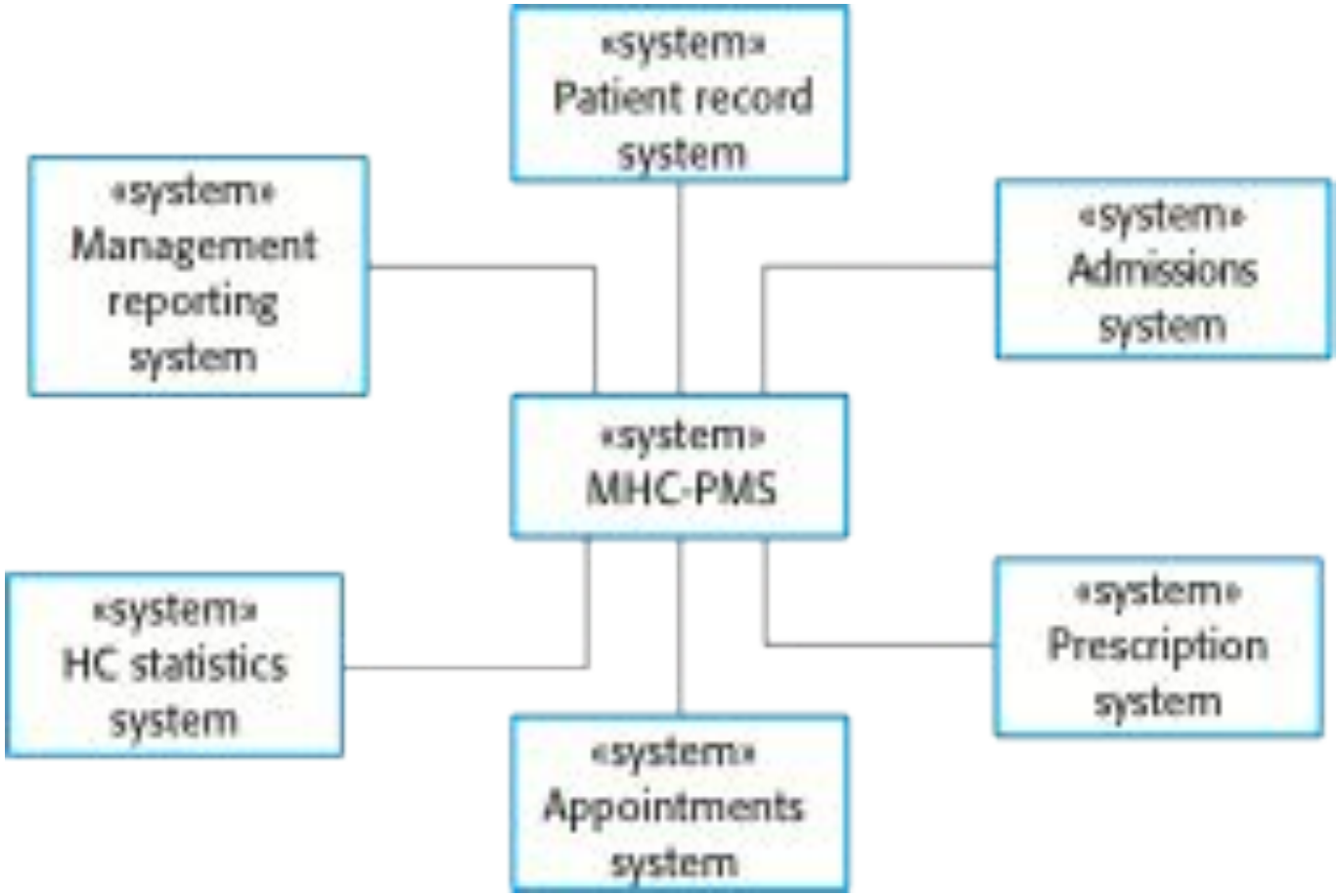
Context models

- Context models are used to illustrate the boundaries of a system
- Architectural models show the a system and its relationship with other systems
  - Context models simply show the other systems in the environment, not how the system being developed is used in that environment.

The context of an ATM system



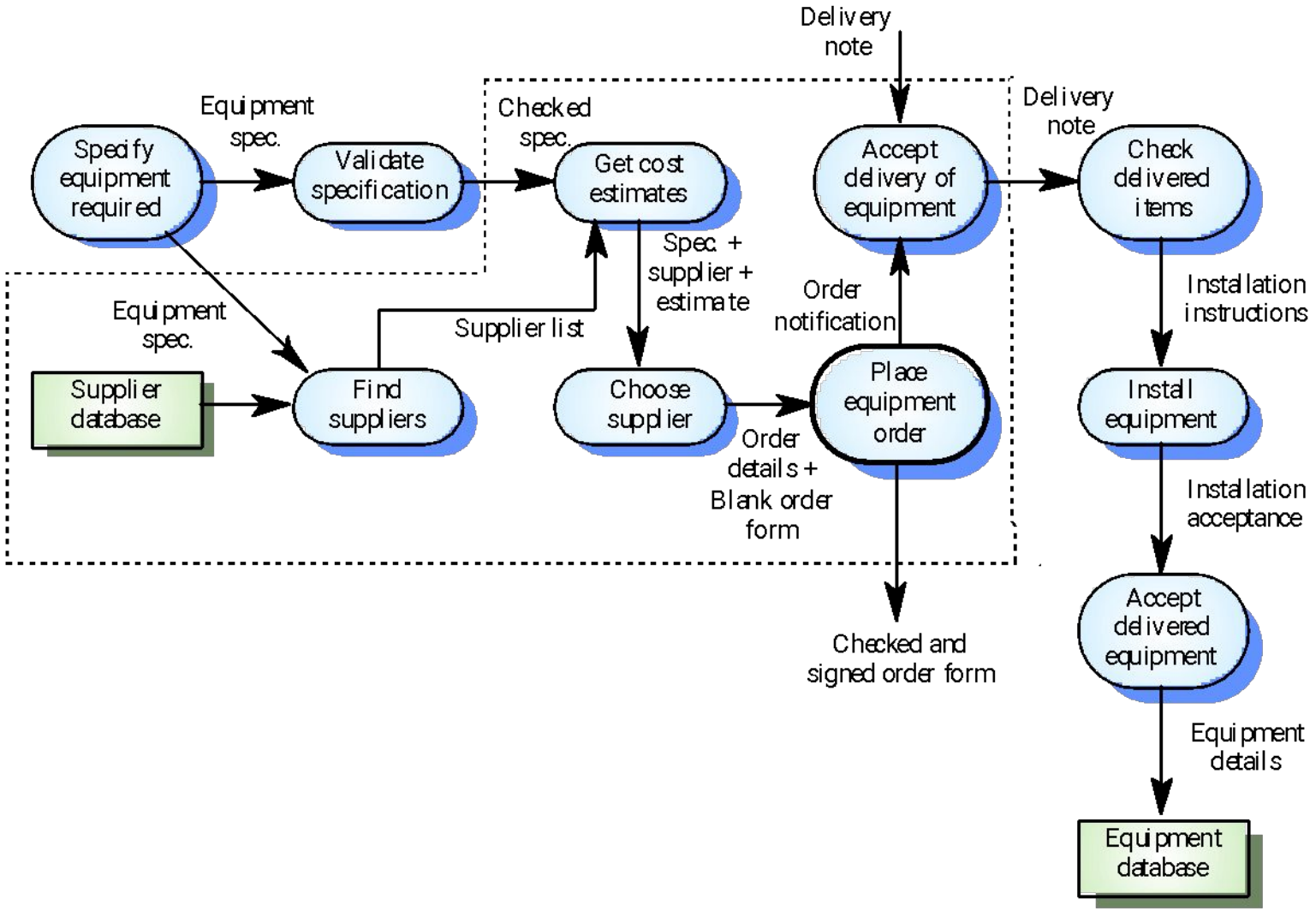
- Architectural models describe the environment of a system. However, they do not show the relationships between the other systems in the environment
- External systems might produce data for or consume data from the system


«system»
Patient record system
«system»
Management reporting system
«system»
Admissions system
«system»
MHC-PMS
«system»
HC statistics system
«system»
Prescription system
«system»
Appointments system

Process models

- simple architectural models are normally supplemented by other models, such as process models, that show the process activities supported by the system.
- Process models show the overall process and the processes that are supported by the system
- Data flow models may be used to show the processes and the flow of information from one process to another

Equipment procurement process



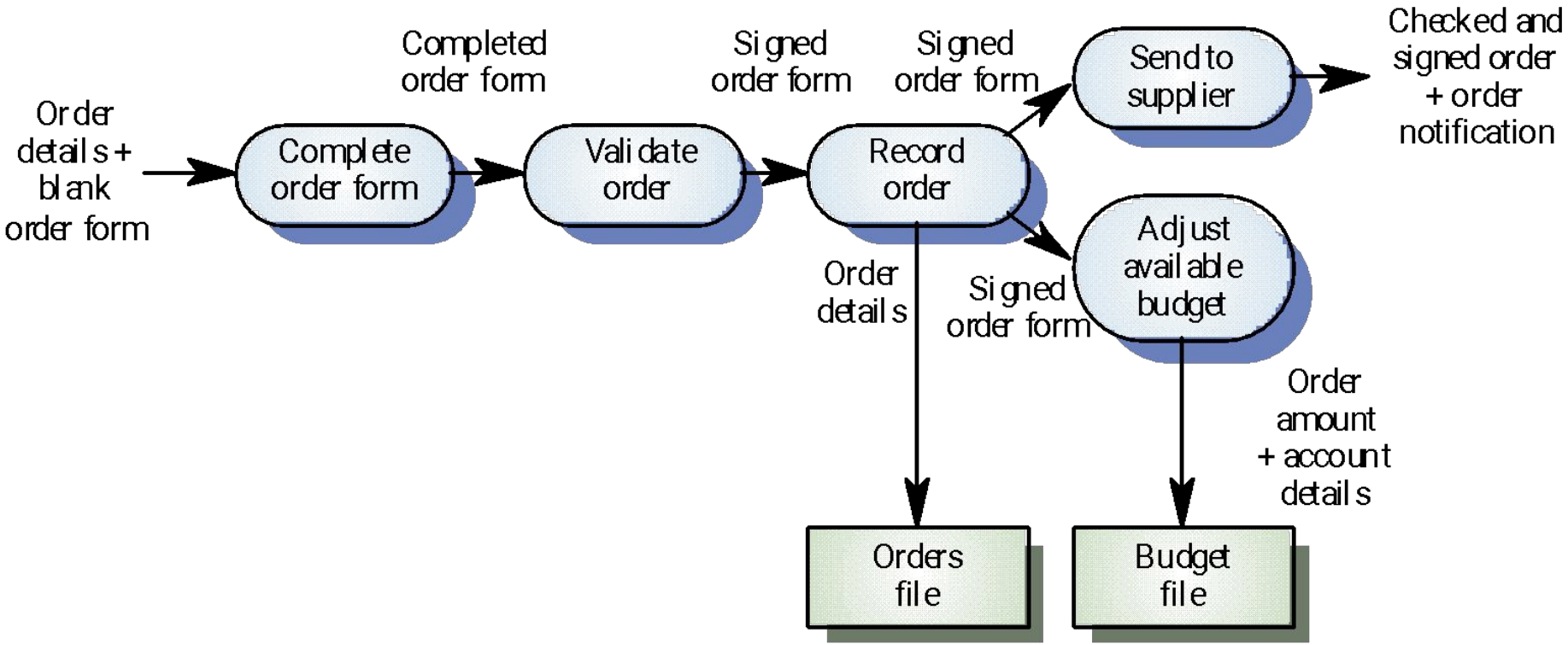
Behavioural models

- Behavioural models are used to describe the overall behaviour of a system
- Two types of behavioural model are shown here
  - **Data processing models** that show how data is processed as it moves through the system
  - **State machine models** that show the systems response to events
- Both of these models are required for a description of the system's behaviour

Data-processing models

- Data flow diagrams are used to model the system's data processing
- These show the processing steps as data flows through a system
- Intrinsic part of many analysis methods
- Simple and intuitive notation that customers can understand
- Show end-to-end processing of data

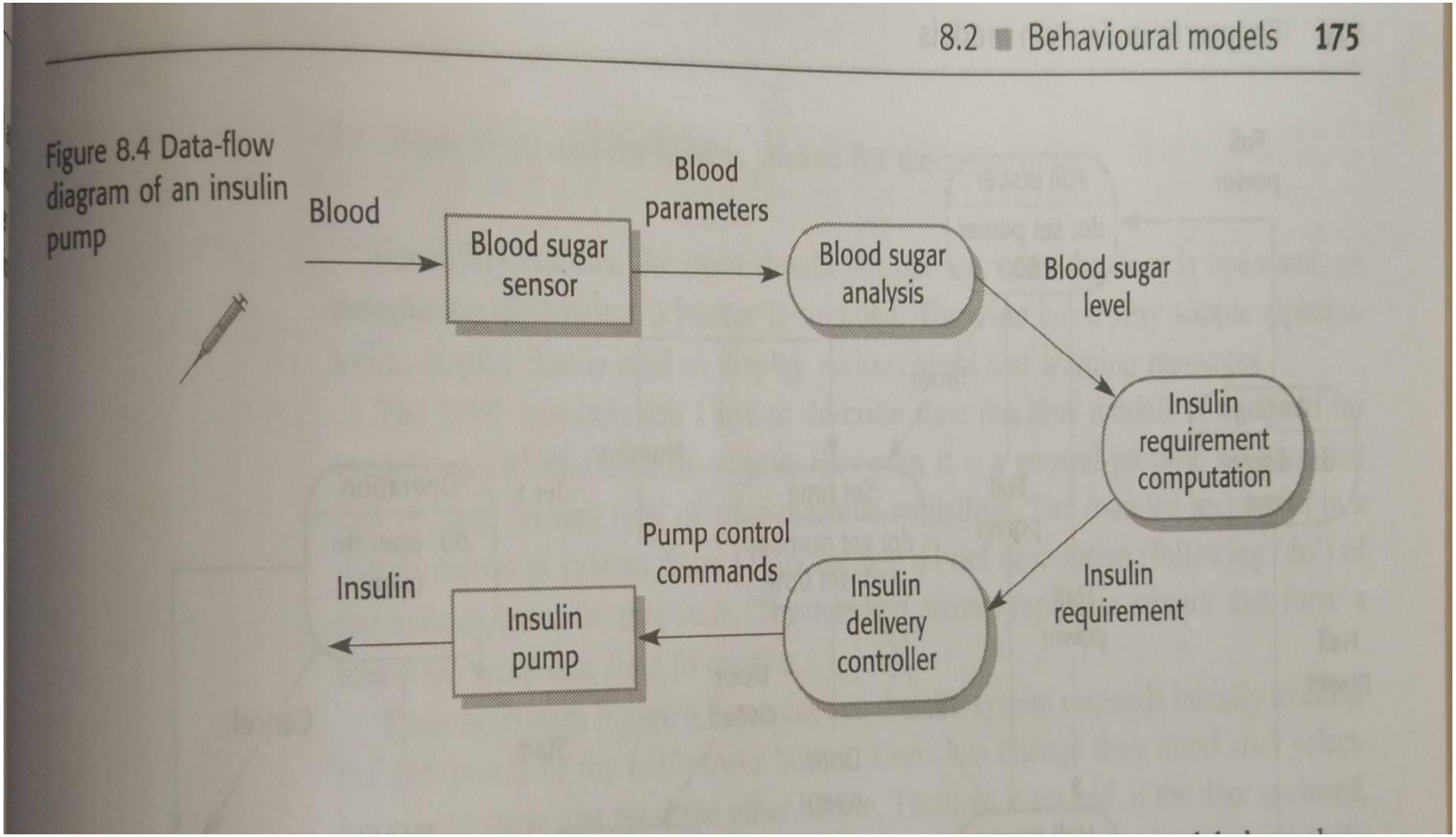
Order processing DFD



Data flow diagrams

- DFDs model the system from a functional perspective
- Tracking and documenting how the data associated with a process is helpful to develop an overall understanding of the system
- Data flow diagrams may also be used in showing the data exchange between a system and other systems in its environment

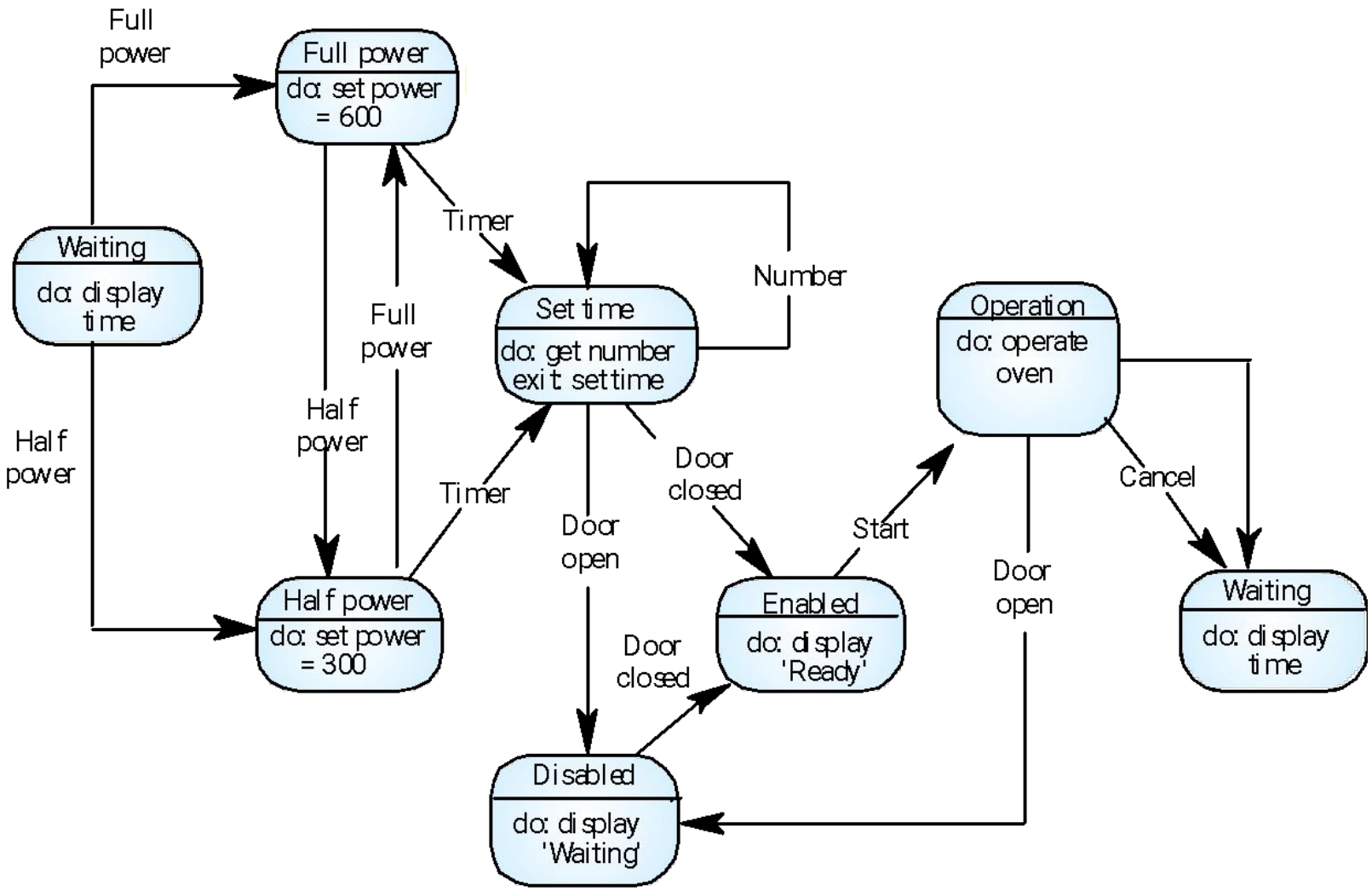
CASE toolset DFD



State machine models

- These model the behaviour of the system in response to external and internal events
- They show the system's responses to stimuli so are often used for modelling real-time systems
- State machine models show system states as nodes and events as arcs between these nodes. When an event occurs, the system moves from one state to another
- Statecharts are an integral part of the UML

Microwave oven model



Microwave oven state description

State	Description
Waiting	The oven is waiting for input. The display shows the current time.
Half power	The oven power is set to 300 watts. The display shows 'Half power'.
Full power	The oven power is set to 600 watts. The display shows 'Full power'.
Set time	The cooking time is set to the user's input value. The display shows the cooking time selected and is updated as the time is set.
Disabled	Oven operation is disabled for safety. Interior oven light is on. Display shows 'Not ready'.
Enabled	Oven operation is enabled. Interior oven light is off. Display shows 'Ready to cook'.
Operation	Oven in operation. Interior oven light is on. Display shows the timer countdown. On completion of cooking, the buzzer is sounded for 5 seconds. Oven light is on. Display shows 'Cooking complete' while buzzer is sounding.

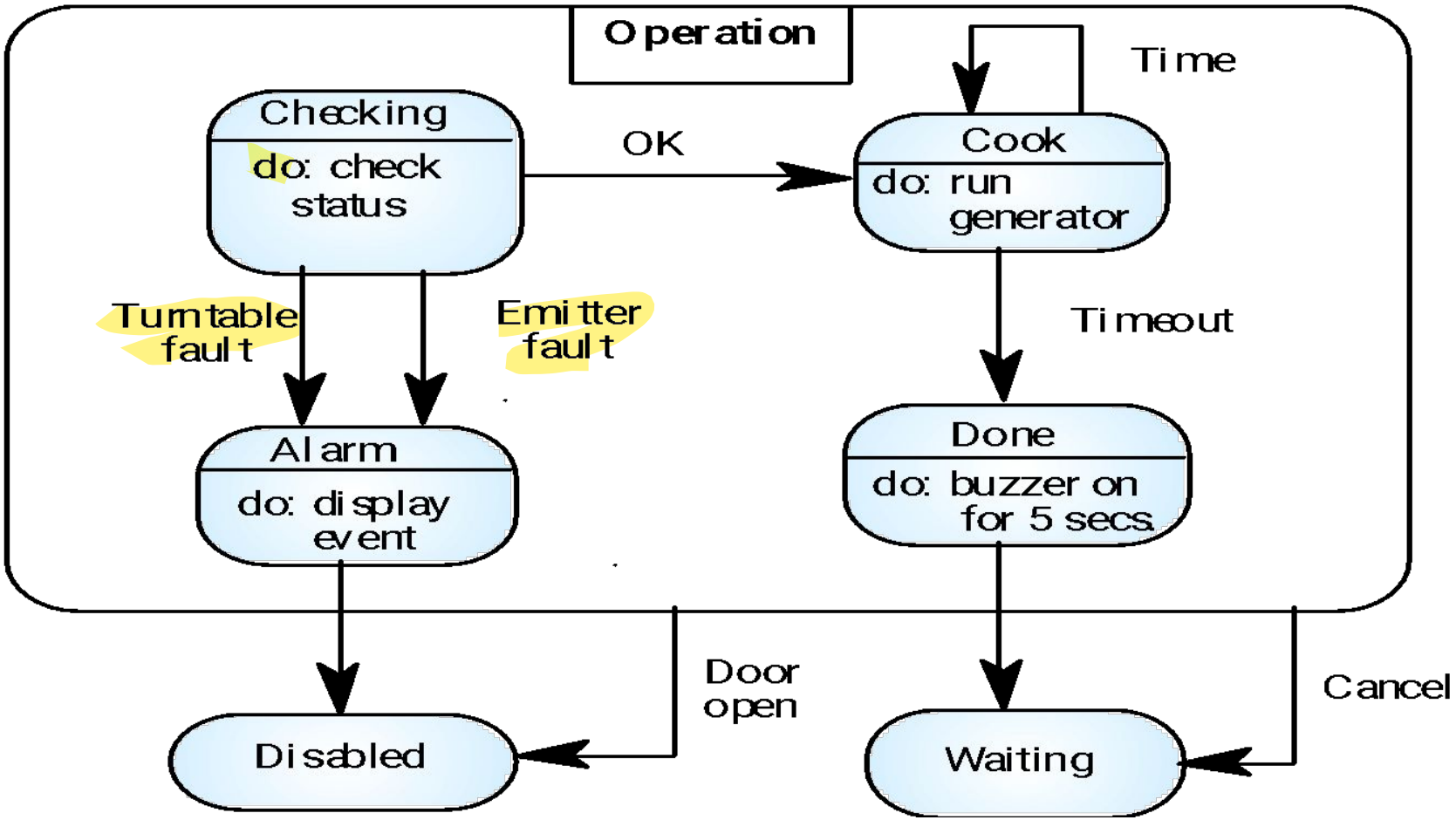
Microwave oven stimuli

Stimulus	Description
Half power	The user has pressed the half power button
Full power	The user has pressed the full power button
Timer	The user has pressed one of the timer buttons
Number	The user has pressed a numeric key
Door open	The oven door switch is not closed
Door closed	The oven door switch is closed
Start	The user has pressed the start button
Cancel	The user has pressed the cancel button

Statecharts

- Allow the decomposition of a model into sub-models (see following slide)
- A brief description of the actions is included following the 'do' in each state
- Can be complemented by tables describing the states and the stimuli

Microwave oven operation



Semantic data models

- Used to describe the logical structure of data processed by the system
- Entity-relation-attribute model sets out the entities in the system, the relationships between these entities and the entity attributes
- Widely used in database design. Can readily be implemented using relational databases
- No specific notation provided in the UML but objects and associations can be used

Software design semantic model

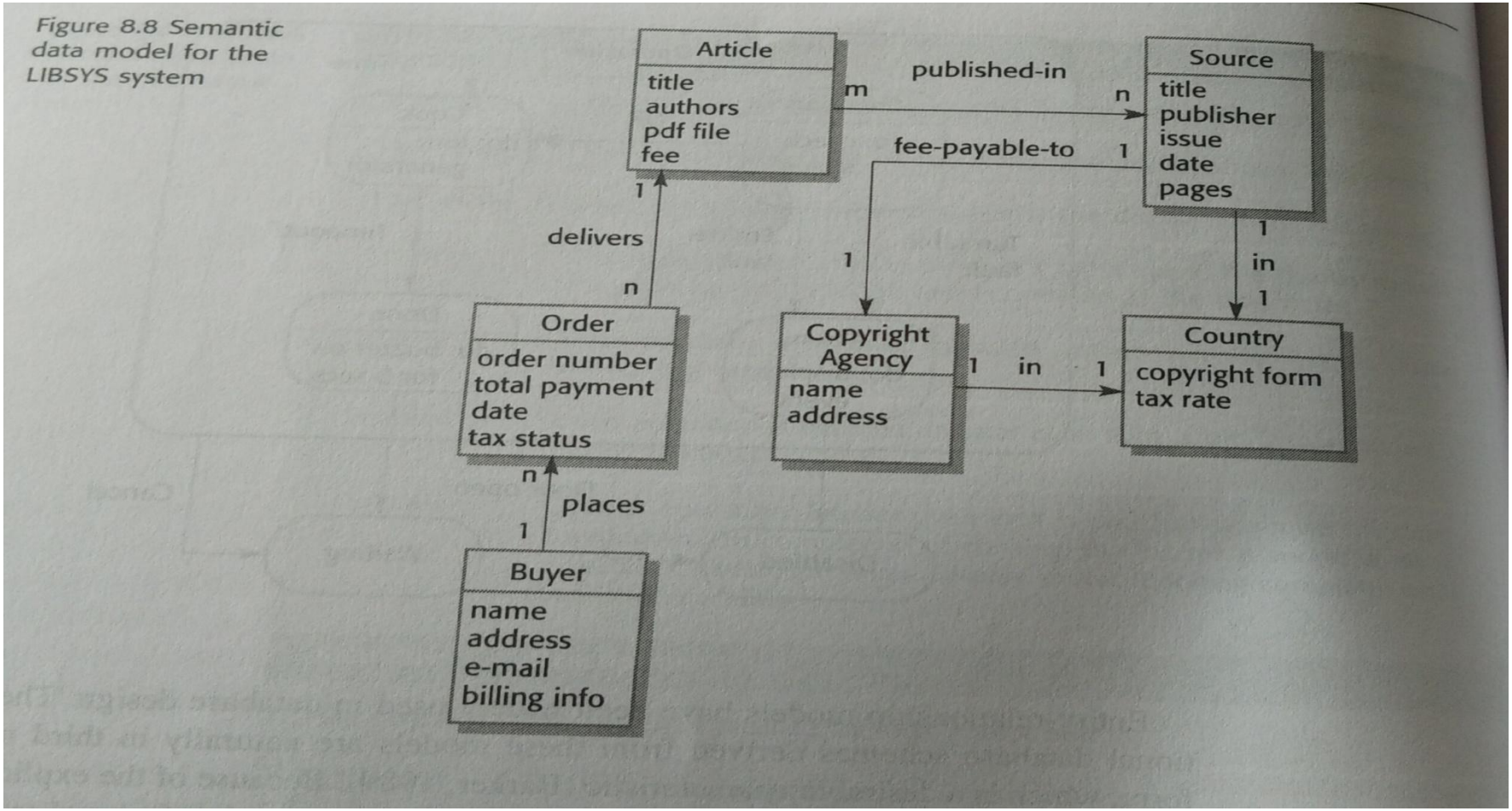


Figure 8.8 Semantic data model for the LIBSYS system

Data dictionaries

- Data dictionaries are lists of all of the names used in the system models. Descriptions of the entities, relationships and attributes are also included
- Advantages
  - Support name management and avoid duplication
  - Store of organisational knowledge linking analysis, design and implementation
- Many CASE workbenches support data dictionaries

Data dictionary entries

Figure 8.9 Examples of data dictionary entries

Name	Description	Type	Date
Article	Details of the published article that may be ordered by people using LIBSYS.	Entity	30.12.2002
authors	The names of the authors of the article who may be due a share of the fee.	Attribute	30.12.2002
Buyer	The person or organisation that orders a copy of the article.	Entity	30.12.2002
fee-payable-to	A 1:1 relationship between Article and the Copyright Agency who should be paid the copyright fee.	Relation	29.12.2002
Address (Buyer)	The address of the buyer. This is used to any paper billing information that is required.	Attribute	31.12.2002

Object models

- Object models describe the system in terms of object classes
- An object class is an abstraction over a set of objects with common attributes and the services (operations) provided by each object
- Various object models may be produced
  - Inheritance models
  - Aggregation models
  - Behaviour models

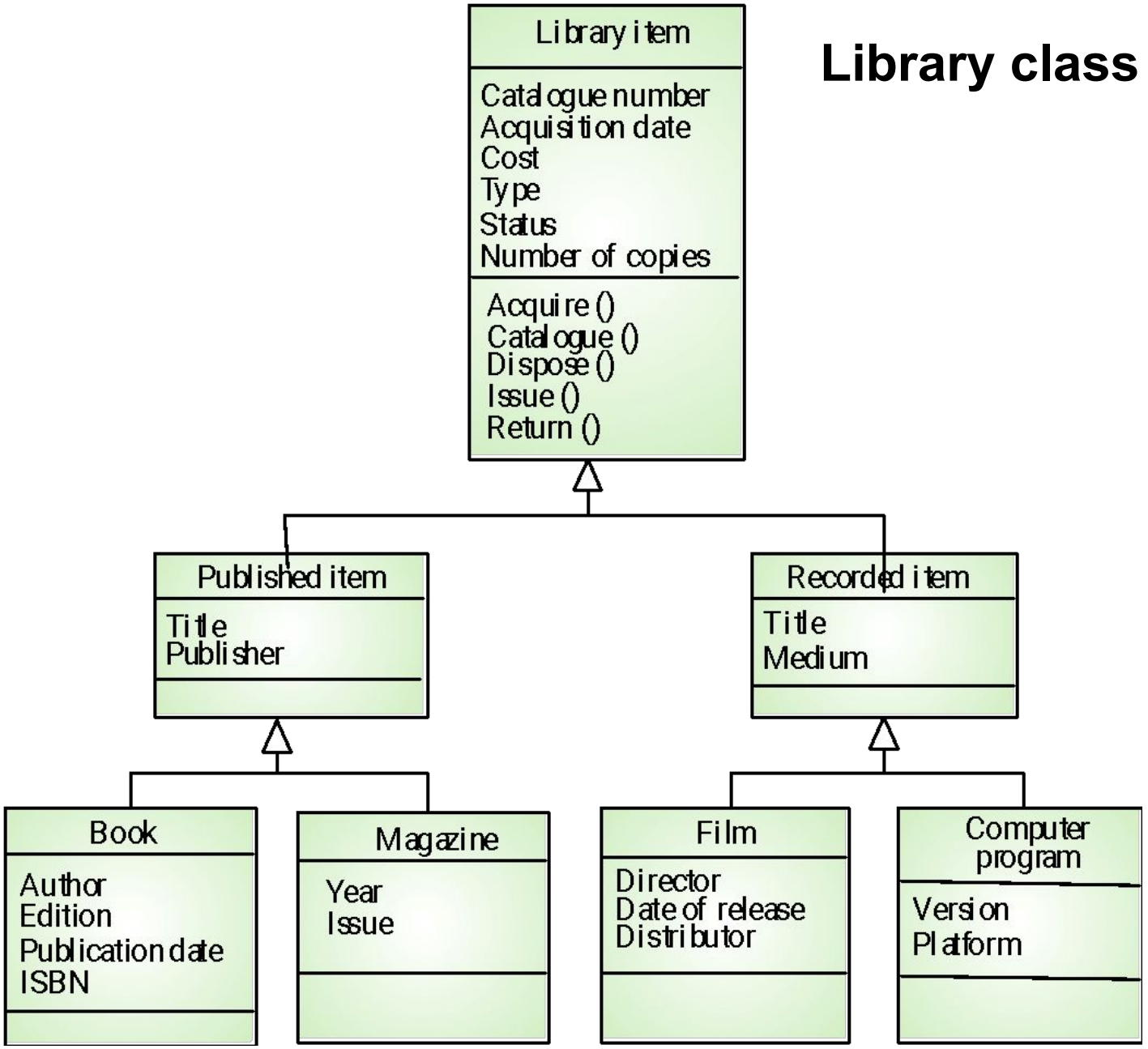
Inheritance models

- Organise the domain object classes into a hierarchy
- Classes at the top of the hierarchy reflect the common features of all classes
- Object classes inherit their attributes and services from one or more super-classes. these may then be specialised as necessary
- Class hierarchy design is a difficult process if duplication in different branches is to be avoided

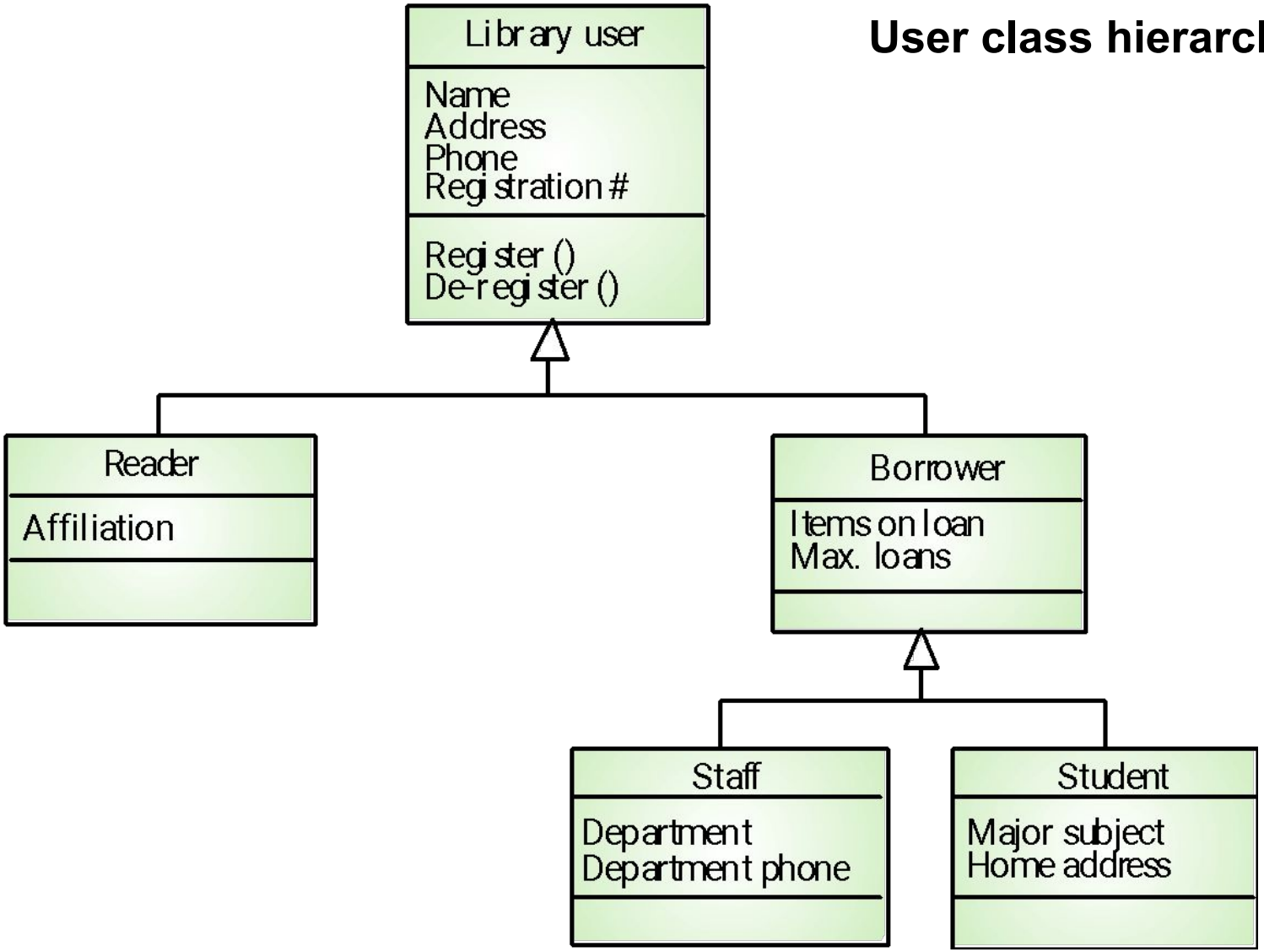
The Unified Modeling Language

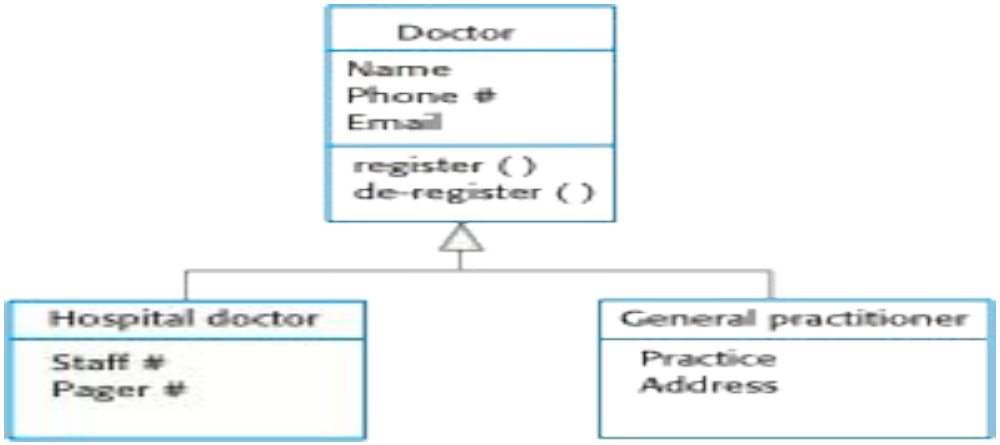
- Devised by the developers of widely used object-oriented analysis and design methods
- Has become an effective standard for object-oriented modelling
- Notation
  - Object classes are rectangles with the name at the top, attributes in the middle section and operations in the bottom section
  - Relationships between object classes (known as associations) are shown as lines linking objects
  - Inheritance is referred to as generalisation and is shown 'upwards' rather than 'downwards' in a hierarchy

Library class hierarchy



User class hierarchy

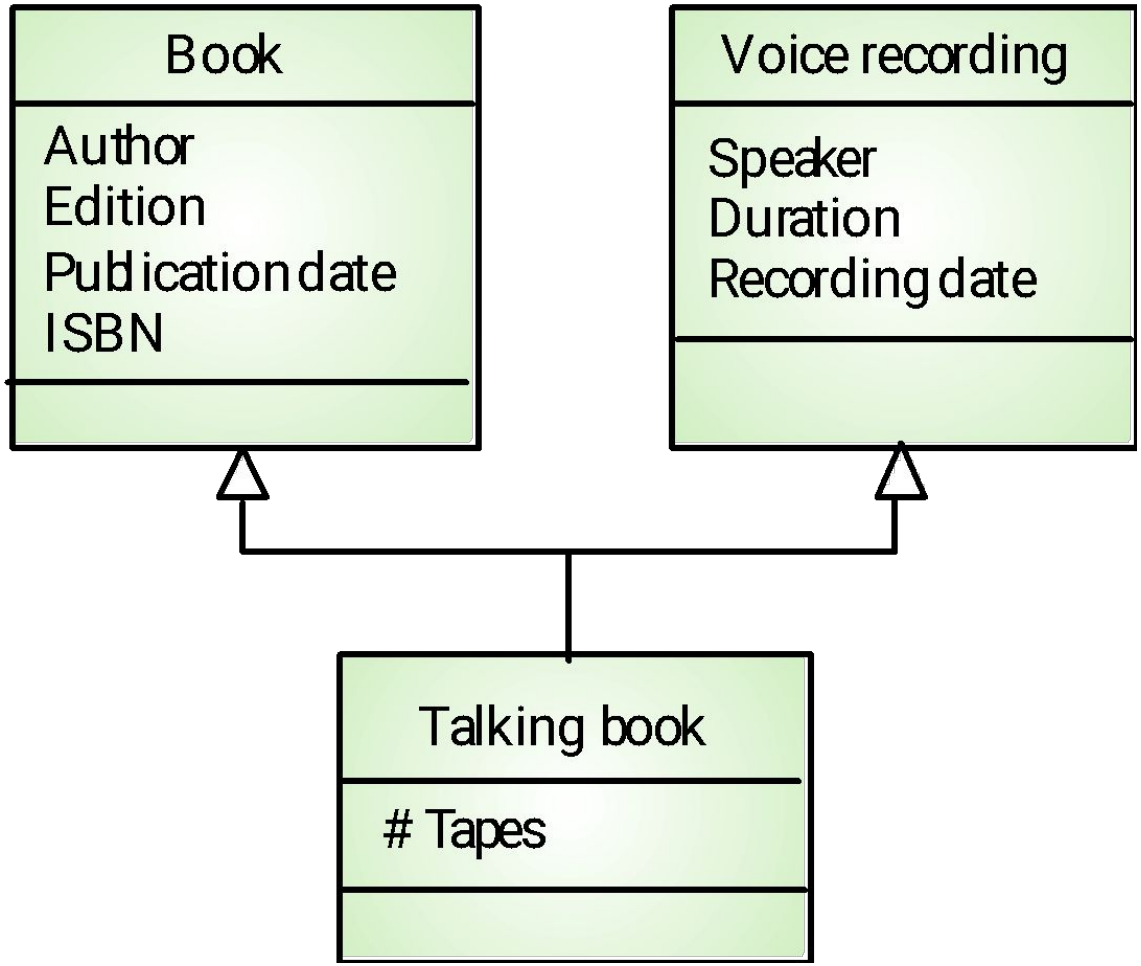



Doctor
Name
Phone #
Email
register ( )
de-register ( )
Hospital doctor
Staff #
Pager #
General practitioner
Practice
Address

Multiple inheritance

- Rather than inheriting the attributes and services from a single parent class, a system which supports multiple inheritance allows object classes to inherit from several super-classes
- Can lead to semantic conflicts where attributes/services with the same name in different super-classes have different semantics
- Makes class hierarchy reorganisation more complex

Multiple inheritance

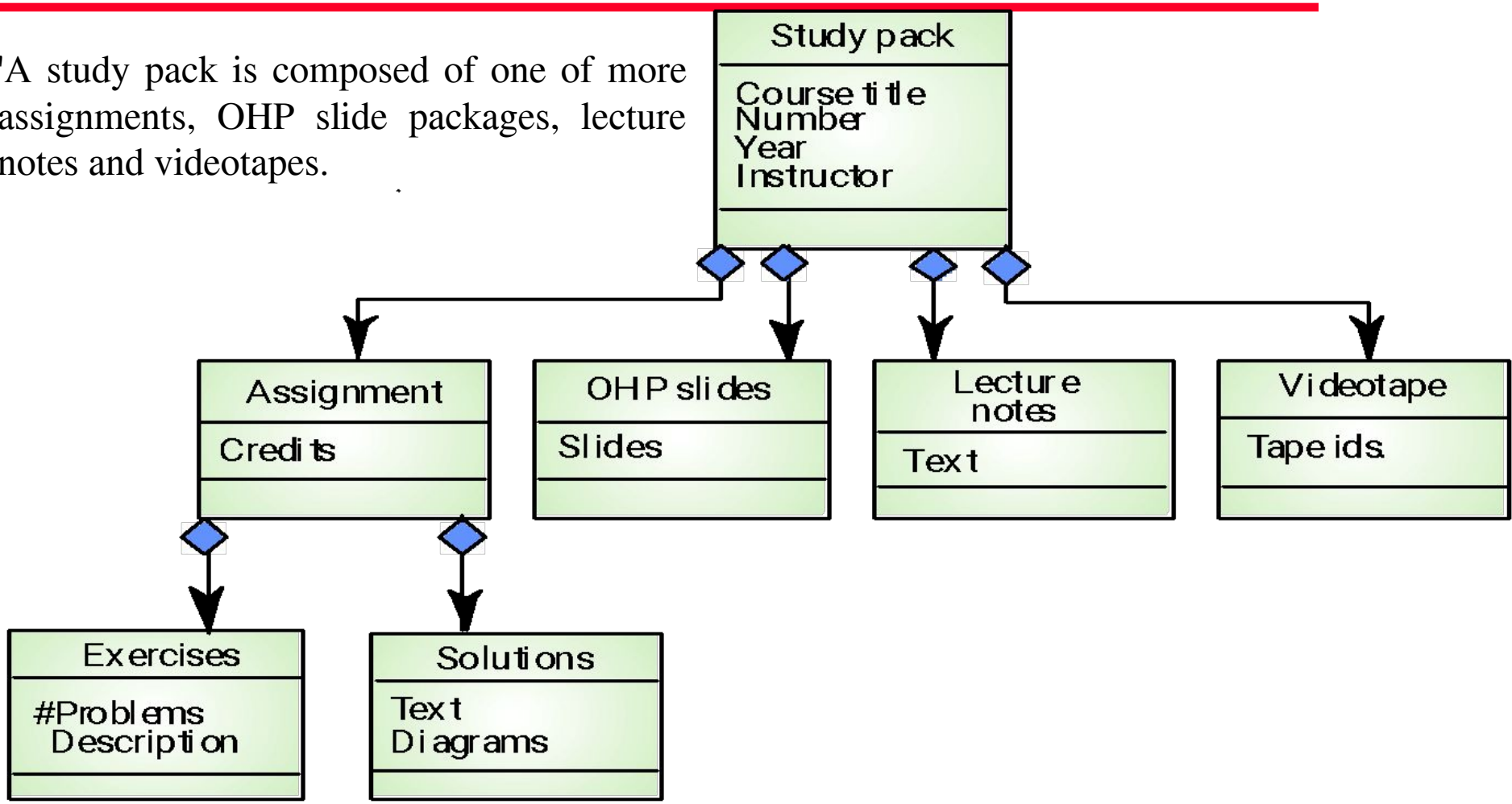


Object aggregation

- Aggregation model shows how classes which are collections are composed of other classes
- Similar to the part-of relationship in semantic data models

Object aggregation

'A study pack is composed of one of more assignments, OHP slide packages, lecture notes and videotapes.

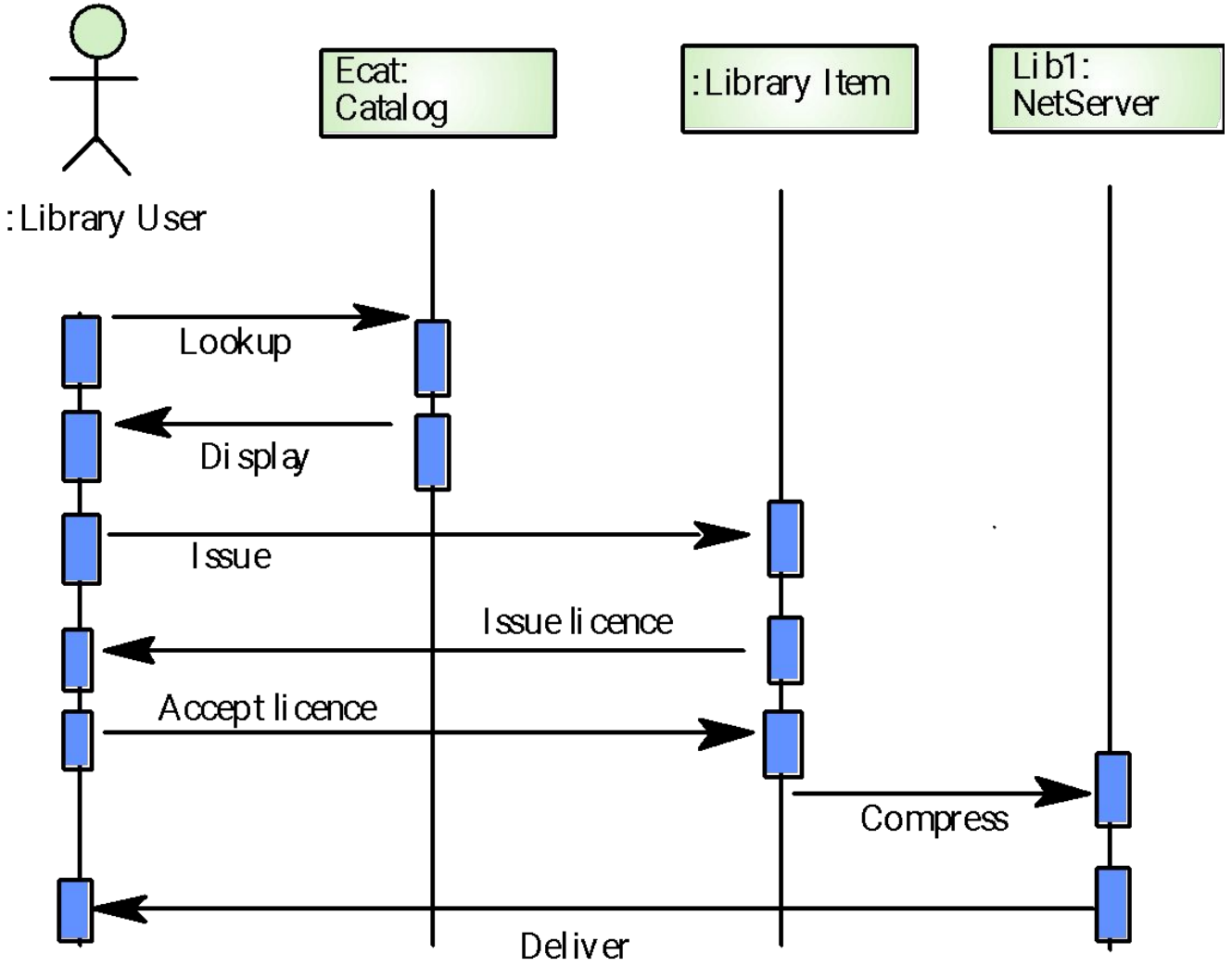


Object behaviour modelling

- A behavioural model shows the interactions between objects to produce some particular system behaviour that is specified as a use-case
- Sequence diagrams (or collaboration diagrams) in the UML are used to model interaction between objects

- where users withdraw items from the library in electronic form.
- In a sequence diagram, objects and actors are aligned along the top of the diagram. Labelled arrows indicate operations; the sequence of operations is from top to bottom.
- In this scenario, the library user accesses the catalogue to see whether the item required is available electronically; if it is, the user requests the electronic issue of that item. For copyright reasons, this must be licensed so there is a transaction between the item and the user where the license is agreed. The item to be issued is then sent to a network server object for compression before being sent to the library user.

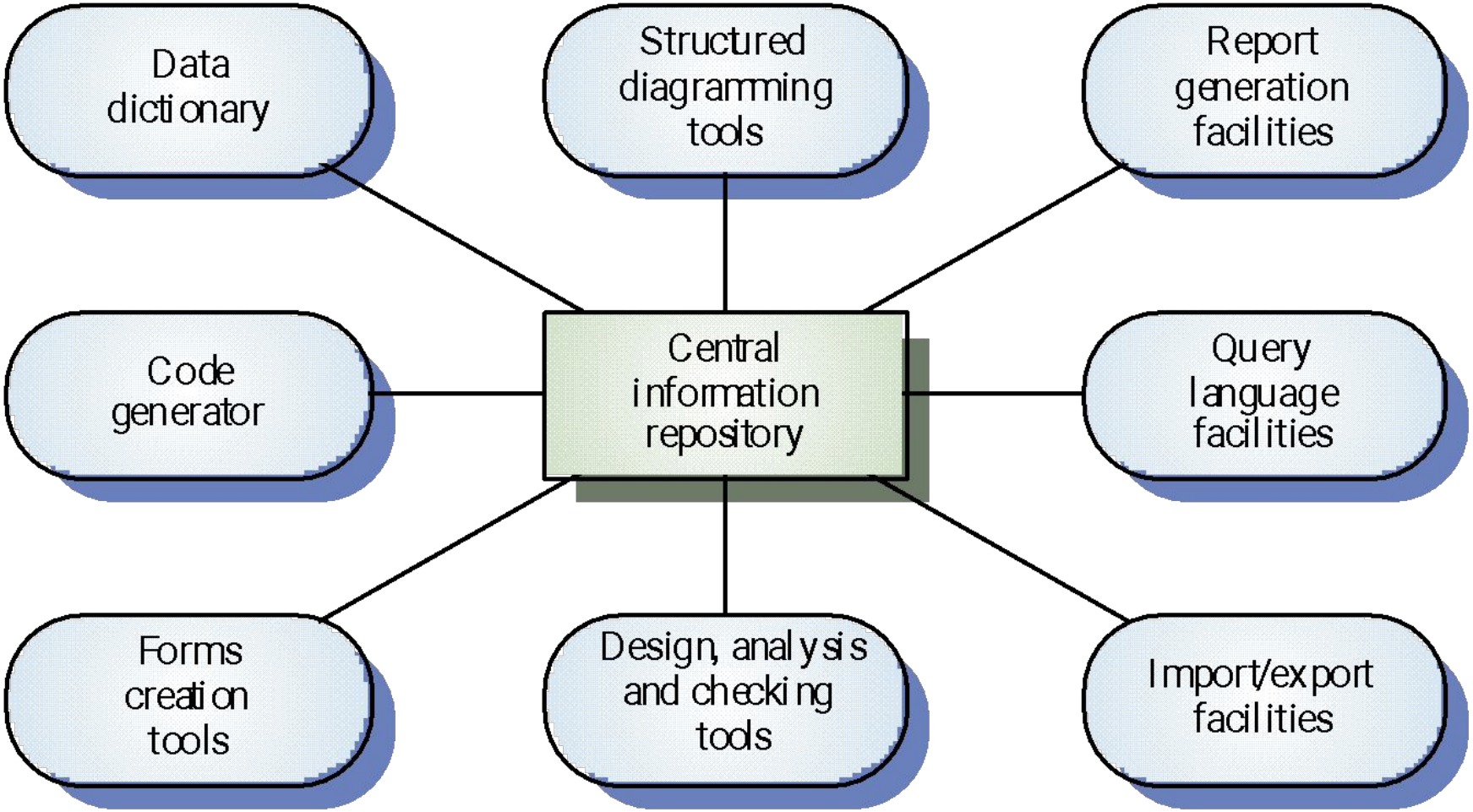
Issue of electronic items



Structured methods

- A structured method is a systematic way of producing models of an existing system or of a system that is to be built.
- Structured methods have been applied successfully in many large projects. They can deliver significant cost reductions because they use standard notations and ensure that standard design documentation is produced.
- A coherent set of tools that is designed to support related software process activities such as analysis, design or testing
- Structured methods provide a framework for supporting the development of system models. They normally have extensive case tool support, including model editing and checking and code generation

An analysis and design workbench



Analysis workbench components

- Diagram editors
- Design analysis and checking tools
- Repository and associated query language
- Data dictionary
- Report definition and generation tools
- Forms definition tools
- Import/export translators
- Code generation tools

- Diagram editors used to create object models, data models, behavioural models, and so on. These editors are not just drawing tools but are aware of the types of entities in the diagram. They capture information about these entities and save this information in the central
- Design analysis and checking tools that process the design and report on error and anomalies. These may be integrated with the editing system so that user errors are trapped at an early stage in the process.
- Repository query languages that allow the designer to find designs and associated design information in the repository.
- A data dictionary that maintains information about the entities used in a system design.
- 5. Report definition and generation tools that take information from the central store and automatically generate system documentation.

- Import/export facilities that allow the interchange of information from the central repository with other development tools.
- Code generators that generate code or code skeletons automatically from the design captured in the central store.
- CASE tools often support different languages so the user can generate a program in C, C++ or Java from the same design model

Key points

- A model is an abstract system view. Complementary types of model provide different system information
- Context models show the position of a system in its environment with other systems and processes
- Data flow models may be used to model the data processing in a system
- State machine models model the system's behaviour in response to internal or external events

Key points

- Semantic data models describe the logical structure of data which is imported to or exported by the systems
- Object models describe logical system entities, their classification and aggregation
- Object models describe the logical system entities and their classification and aggregation
- CASE workbenches support the development of system models

MCQ's on Module2

1.Which model in system modelling depicts the dynamic behaviour of the system ?
a) Context Model
b) Behavioral Model
c) Data Model
d) Object Model

2. Which model in system modelling depicts the static nature of the system ?
a) Behavioral Model
b) Context Model
c) Data Model
d) Structural Model

(Explanation: Structural models show the organization and architecture of a system. These are used to define the static structure of classes in a system and their associations.)

3.Which perspective in system modelling shows the system or data architecture.

a) Structural perspective
b) Behavioral perspective
c) External perspective
d) All of the mentioned

4. Which system model shows what lies outside the system boundaries.

a) Structural model
b) Context model
c) Behavioral model
d) Interaction model

5. ___State__diagrams show system states and events that cause transitions from one state to another.

6. _____DFDs__ focus on system functions and do not recognize system objects.

7._____allows us to infer that different members of classes have some common characteristics

a. Generalization

b. Aggregation

C. Dependency

8. Directed arc or line in DFD represents

A. Data Store
B. Data Process
C. Data Flow
D. All of the above

8.

9.Which of the following is a function of CASE Tool?
A. Supporting Structured analysis and design (SA/SD)
B. Maintaining the data dictionary
C. Checking whether DFDs are balanced or not
D. None of the above

- View Answer
- Ans : A

Explanation: Supporting Structured analysis and design (SA/SD) is a function of CASE Tool.

9.

10. Which diagram of UML represent Interaction modeling?

A. Use Case

B. Sequence

C. State Chart

D. Both A and B

View Answer

- Ans : D

Explanation: Use case modeling is mostly used to model interactions between a system and external actors. Sequence diagrams are used to model interactions between system components, although external agents may also be included.

10. A rectangle in a DFD represents

a process	
b. an external entity	
c. an input unit	
d. a data store	

- In a DFD external entities are represented by a (a) rectangle
- (b) ellipse
- (c) diamond shaped box
- (d) circle