System models

- Abstract descriptions of systems whose requirements are being analysed

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

Structured methods

- Structured methods incorporate system modelling as an inherent part of the method
- Methods define a set of models, a process for deriving these models and rules and guidelines that should apply to the models
- CASE tools support system modelling as part of a structured method
Method weaknesses

- They do not model non-functional system requirements
- They do not usually include information about whether a method is appropriate for a given problem
- They may produce too much documentation
- The system models are sometimes too detailed and difficult for users to understand

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
- 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
- Social and organisational concerns may affect the decision on where to position system boundaries
- Architectural models show the system and its relationship with other systems
The context of an ATM system

- Security system
- Account database
- Auto-teller system
- Usage database
- Maintenance system
- Branch accounting system
- Branch counter system

Process models

- 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

- Get cost estimates
- Accept delivery of equipment
- Check delivered items
- Validate specification
- Specify equipment required
- Choose supplier
- Place equipment order
- Install equipment
- Find suppliers
- Supplier database
- Accept delivered equipment
- Equipment database
- Equipment spec. checked
- Spec. + supplier + estimate
- Supplier list
- Order notification
- Installation instructions
- Installation acceptance
- Equipment details
- Equipment details
Behavoural 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 system's 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**

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

**Microwave oven stimuli**

<table>
<thead>
<tr>
<th>Stimulus</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Half power</td>
<td>The user has pressed the half power button</td>
</tr>
<tr>
<td>Full power</td>
<td>The user has pressed the full power button</td>
</tr>
<tr>
<td>Timer</td>
<td>The user has pressed one of the timer buttons</td>
</tr>
<tr>
<td>Number</td>
<td>The user has pressed a numeric key</td>
</tr>
<tr>
<td>Door open</td>
<td>The oven door switch is not closed</td>
</tr>
<tr>
<td>Door closed</td>
<td>The oven door switch is closed</td>
</tr>
<tr>
<td>Start</td>
<td>The user has pressed the start button</td>
</tr>
<tr>
<td>Cancel</td>
<td>The user has pressed the cancel button</td>
</tr>
</tbody>
</table>
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 relationships between 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
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
  - Interaction models

Object models

- Natural ways of reflecting the real-world entities manipulated by the system
- More abstract entities are more difficult to model using this approach
- Object class identification is recognised as a difficult process requiring a deep understanding of the application domain
- Object classes reflecting domain entities are reusable across systems
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

- **Library item**
  - **Catalogue number**
  - **Acquisition date**
  - **Type**
  - **Status**
  - **Number of copies**
  - **Acquire()**
  - **Catalogue()**
  - **Dispose()**
  - **Issue()**
  - **Return()**

- **Published item**
  - **Title**
  - **Author**
  - **Edition**
  - **Publication date**
  - **ISBN**

- **Recorded item**
  - **Title**
  - **Director**
  - **Date of release**
  - **Distributor**
  - **Version**
  - **Platform**

- **Magazine**
  - **Year**

- **Book**
  - **Publisher**

- **Film**
  - **Director**
  - **Date of release**

- **Computer program**
  - **Version**
  - **Platform**
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

- Tapes
  - Talking book
    - Author
    - Edition
    - Publication date
    - ISBN
  - Book
    - ISBN
  - Voice recording
    - Speaker
    - Duration
    - Recording date
  - Voice recording

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

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
CASE workbenches

- A coherent set of tools that is designed to support related software process activities such as analysis, design or testing
- Analysis and design workbenches support system modelling during both requirements engineering and system design
- These workbenches may support a specific design method or may provide support for creating several different types of system model

An analysis and design workbench
## Analysis workbench components

- Diagram editors
- Model 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