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
- CASE workbenches

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-flow model: 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.

Models vs perspectives

1. ***Context*** models (external perspective)
2. ***Behavioural*** models (behavioural perspective)
3. ***Data*** models (structural perspective)
4. ***Object*** models (structural perspective)

Context models

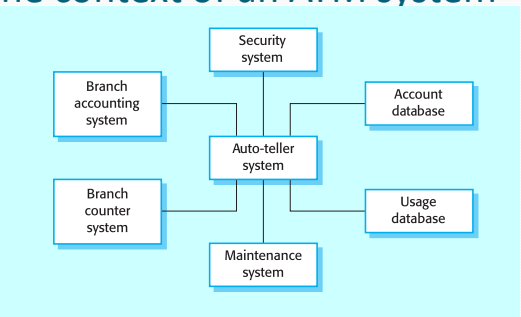
- Context models are used to illustrate the ***operational context of a system*** - they show what lies outside the system boundaries.
- Social and organisational concerns may affect the decision on where to position system boundaries.
- They show the system and ***its relationship with other systems***.

Components and functionalities

Two dimensions for bounding:

- ***Who*** does something (components)
- ***What*** has to be done (functionalities)

The context of an ATM system

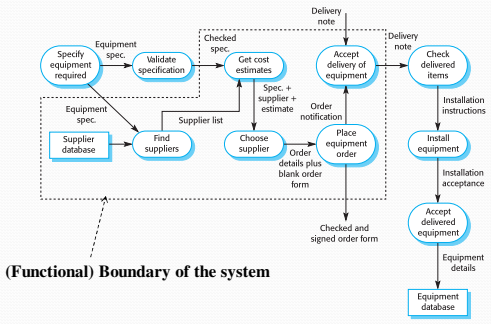


...a component wise

Functionality-wise context: 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



Type of models

1. ***Context*** models
   (data processing, architectural)
2. ***Behavioural*** models
3. ***Data*** models
4. ***Object*** models

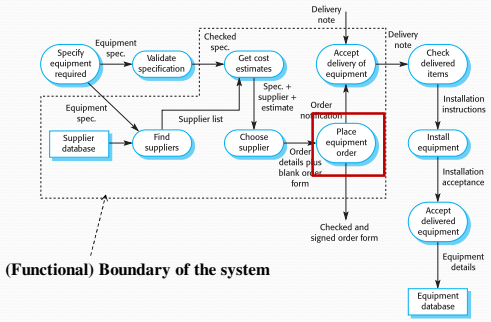
Behavioural models

- Behavioural models are used to describe the overall behaviour of a system.
- Two types of behavioural model are:
  - ***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.
- These models show different perspectives so both of them ay be required to describe the system's behaviour.

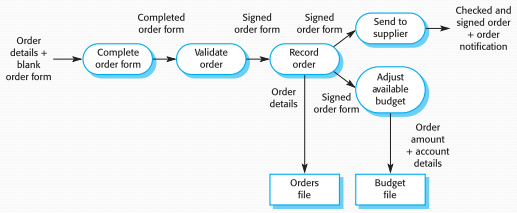
Data-processing models

- Data flow diagrams (DFDs) may be used to model the system's data processing.
- These show the processing steps as data flows through a system.
- DFDs are an intrinsic part of many analysis methods.
- Simple and intuitive notation that customers can understand.
- Show end-to-end processing of data.
- They support hierarchical modeling: from coarse grain to fine grain.

Equipment procurement process



Order processing DFD

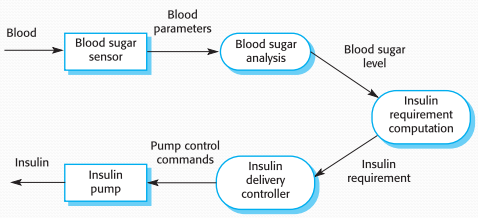


Different granularity: this DFD describe the Place Equipment Order activity of the previous DFD
Rounded blocks: system tasks or functionalities
Rectangles: data stores (data source sor destinations)

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(course grain blocks).

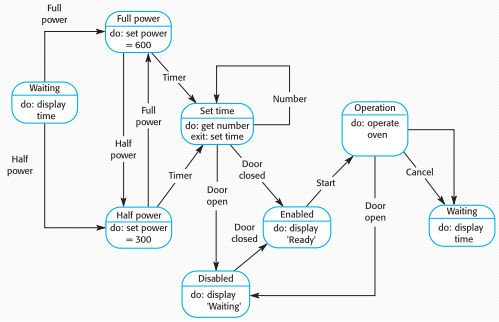
Insulin pump DFD



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 and are used to represent state machine models.

Microwave oven model



Statecharts

- 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.
- Advantage: the number of possible states increasies rapidly!
  - Statecharts allow the decomposition of a model into sub-models.

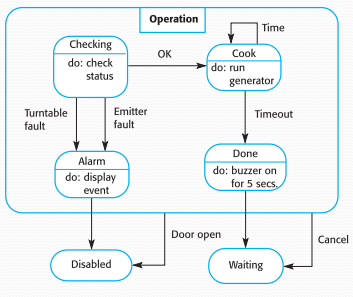
Microwave oven state description

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 ÔHalf powerÕ
Full power	The oven power is set to 600 watts. The display shows ÔFull powerÕ
Set time	The cooking time is set to the userÕs 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 ÔNot readyÕ
Enabled	Oven operation is enabled. Interior oven light is off. Display shows ÔReady to cookÕ
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 ÔCooking completeÕ 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

Hierarchical modeling: Microwave oven operation



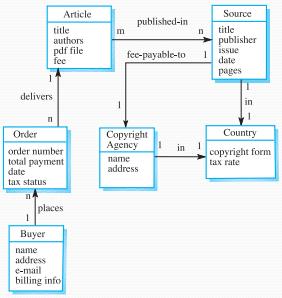
Type of models

1. ***Context*** models
   (data processing model, architectural model)
2. ***Behavioural*** models
   (data processing model, state machine model)
3. ***Data*** models
4. ***Object*** models

Semantic **data models**

- Used to describe the ***logical structure of data*** processed by the system.
- An ***Entity-Relation-Attribute*** (ERA) 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.

Library semantic model



This is not an ERA model, but rather a (incomplete) **class diagram**

*Try to translate this diagram into an **ERA model***

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).
  - Other information: description of compositions, date of creation, the creator and the representation used.
- 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

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 co py 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

Type of models

1. ***Context*** models
   (data processing model, architectural model)
2. ***Behavioural*** models
   (data processing model, state machine model)
3. ***Data*** models
   (architectural, compositional and classification models)
1. ***Object*** models

Object models

- Object models describe the ***system*** in terms of ***object classes and their associations***.
- 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 can be a difficult process if duplication in different branches is to be avoided.

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.
- This can lead to semantic conflicts where attributes/services with the same name in different super-classes have different semantics.
- Multiple inheritance makes class hierarchy reorganisation more complex.

Object aggregation

- An aggregation model shows how classes that are collections are composed of other classes.
- Aggregation models are similar to the part-of relationship in semantic data models.

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.

More details on o-o design in the next week and on UML in Lab course

Type of models

1. ***Context*** models
   (data processing model, architectural model)
2. ***Behavioural*** models
   (data processing model, state machine model)
3. ***Data*** models
   (architectural, compositional and classification models)
1. ***Object*** models
   (compositional and classification models)

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.

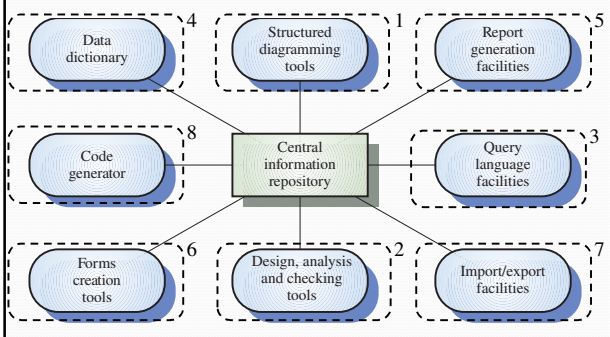
Method weaknesses

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

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 a creating several different types of system model.

An analysis and design workbench…



… and component tasks

1. Diagram editors
2. Process the design and report on errors
3. Find designs and associated information
4. Information about the entities in the design
5. Automatic generation of documentation
6. Specification of screen and document formats
7. Information interchange with other tools
8. Automatic code generation from design

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

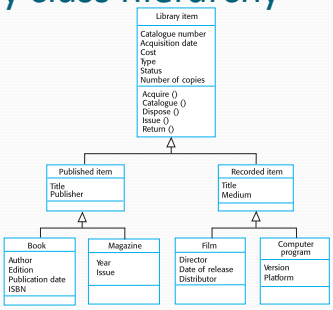
Key points

- 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.
- Sequence models show the interactions between actors and the system objects that they use.
- Structured methods provide a framework for developing system models.

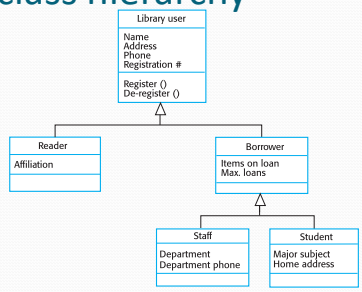
Object models and the UML

- The UML is a standard representation devised by the developers of widely used object-oriented analysis and design methods.
- It 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.

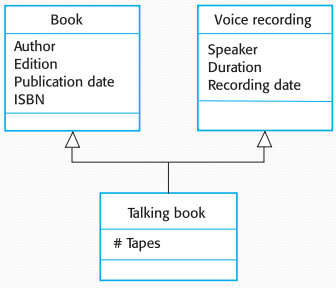
Library class hierarchy



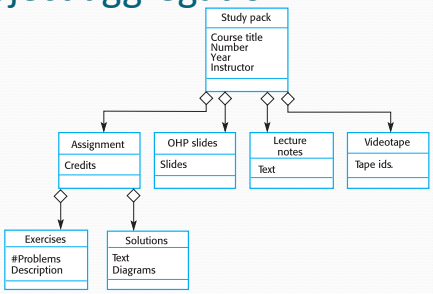
User class hierarchy



Multiple inheritance



Object aggregation



Issue of electronic items