Modelling as a Communication Tool: Introduction to Process Modelling

CSE1204 - Information Systems 1

The requirements specification document
- Must be communicated to key stakeholders
- Should contain:
  - Functions and services the system should provide
  - Non-functional requirements
  - Constraints affecting the system’s development and operations
  - Information about other systems the system must interface with
- System models are used to help understand the existing system and describe the proposed system

Example system
- Easy Go Hotel

Modelling
- Why do we do it?
  - Our own understanding
  - Communication with others
- How do we do it?
  - informal techniques
  - formal techniques
- How effective is it?
  - different techniques for different purposes
  - e.g., a road map, an organisation chart, a data flow diagram

Simplification in modelling
- all models are simplifications of the real world: they omit some features and emphasise others
- this is called “abstraction”
- the choice of model and modelling method requires decisions about:
  - what things should be included
  - what things can be omitted

Representation in modelling
- a model is a simulation: it is composed from something which represents reality

Suitability
- the choice of model and modelling method requires decisions about the form of representation that can best represent the real-life object being modelled
Partitioning and level of detail in modelling

- **Partitioning:** all models simplify complex reality by breaking it down into smaller, less complex parts which can be considered separately.
- Partitioning allows the modeller to vary the amount of detail which is given at different levels of the model.
- The choice of model and modelling method requires decisions about:
  - How to partition without losing the overall picture of the thing being modelled.
  - How much detail can be included at each level of the model.

Audience

- The suitability of different forms and languages for modelling will vary for different audiences.
- All models assume some level of familiarity and understanding on the part of the audience.
- The choice of model and modelling method requires decisions about which modelling form and language the target audience will best understand.

Purpose of models

- All models are built to serve some purpose.
- There is no 'best' model independent of any purpose.
- The quality of a model is dependent on the purpose for which it was built.
- The choice of model and modelling method requires an understanding of the model's purpose and decisions about the best method for achieving that purpose.

Models in analysis and design

- In order to model a system, the systems analyst must choose an appropriate modelling form in terms of its:
  - Audience.
  - Purpose.
  - Degree of abstraction (simplification).
  - Form of representation.
  - Level of partitioning.
  - Level of detail.

Modelling: for whom?

- System users and sponsors:
  - Saltwater Head Librarian; Shire Council; residents of Saltwater.
- Other systems analysts:
  - Analysts who will help build the system.
- Designers:
  - Designers of the finished system.
- Other technical staff - programmers, database builders, systems administrators, etc:
  - Code writers, database designers, help desk, technical support.

What can we use to build a model?

- Words - written descriptions.
- Pictures - photographs, drawings, diagrams, graphs.
- Mixed (words+pictures) - charts, annotated drawings, maps.
- Physical models - real life equivalents, scaled models, simulations.
Model paraphernalia

- standardisation of symbols
- special modelling terminology/ rules

provides a “shorthand” way of conveying the model’s message

What aspects can we model?

- to illustrate the relevant aspects of a system, we must use models of appropriate system components

  - for example, for a house we can model:
    - its appearance (to show the look and feel)
    - the physical layout of rooms (to show functions)
    - the layout and connections between its structural components (to show the builders how to construct it)

- What are the components of an information system which we can use in a model to illustrate specific features of that system?

Modelling information systems

- Three aspects of information systems are typically modelled:
  - data
    - what information is used in the system: entity relationship diagrams, data structure diagrams
  - process
    - what jobs use or manipulate data in the system: function decomposition, structure charts, data flow diagrams
  - behaviour
    - what changes are wrought on information in the system: entity life history diagrams, state transition diagrams

Problems in modelling

- will other people interpret our models of the world as we do?
- what happens to things which we cannot model (exceptions; non-standard processes; errors)?
- can we communicate all aspects of a system satisfactorily using our standard modelling techniques?
- which models work best with which information and which audience?
- standardisation
- generalisation
- abstraction

Process modelling

- processes are the “action” part of businesses

- process modelling graphically represents the processes which act on data to
  - capture
  - manipulate
  - store
  - distribute

Process modelling

- principal techniques
  - function decomposition
  - data flow diagrams

- associated techniques for modelling the details of low-level processes
  - structured English
  - decision tables and decision trees
Data Flow Diagrams

- model the flow of data into, through, and out of an information system
- represent an information system as a network of communicating processes
  - show the processes that change or transform data
  - show the movement of data between processes

Example - Easy Go Hotel

- Make a reservation
- Check in
- Use safe deposit
- Check out

Easy Go Hotel

Components of a DFD

- process
- data flow
- data store
- external agent

Alternative sets of symbols

- process
- data flow
- data store
- external agent

Gane & Sarson
DeMarco
Process
- represents the work performed which changes data
- transforms incoming data flows into outgoing data flows
- has a unique number and name

Naming processes
- name each process using a verb and a noun phrase:
  - "calculate price"
  - "validate customer details"
  - "accept supplier delivery"
- the name of a process should describe what the process does
- avoid vague names such as "process data"
- the number of a process is an identifier .. it does not indicate the sequence of processing

Data Flow
- represents data in motion
- describes a "packet" of data or data that move together
- may consist of many individual, related elements that move together to the same destination

Naming data flows
- name each data flow using a noun or noun phrase
- the name should describe the contents and should include as much information as possible about the data flow

Data Store
- represents a collection of data flows at rest
- has a unique name which should describe the contents of the data store
- may represent many different types of physical locations of data
- may be a temporary or a permanent repository of data
Suppliers

External Agent (source/sink)
- represents an entity in the environment with which the system communicates
- a source if it is an origin of data coming into the system
- a sink if it is a destination for data leaving the system

External Agent (source/sink)
- data flows connecting the external agents to the processes within the system represent the interface between the system and its environment
- external agents are outside the system and define its boundaries
- an external agent may be both a source and a sink
- what a sink does with data it receives from the system and how a source produces data which it inputs to the system are outside the boundary of the system and are not shown on the data flow diagram

Example Data Flow Diagram

Guidelines for Drawing DFDs
- each object on a data flow diagram must have a unique name
- each process must have at least one data flow coming in (input) and at least one data flow going out (output)
- the inputs to a process are different from the outputs of that process
- a process must be able to build its outputs using only the information in its input data flows plus any constant information

Guidelines for Drawing DFDs
- data flows are permitted:
  - between 2 processes
  - from a data store to a process
  - from a process to a data store
  - from a source to a process
  - from a process to a sink
- data flows are NOT permitted:
  - between 2 external agents
  - between 2 data stores
  - from an external agent to a data store

see page 287 in Hoffer, George and Valacich
Guidelines for Drawing DFDs

1. **APPLICATIONS**
   - register application
   - reject application

2. **APPLICATIONS**
   - validate application
   - reject application
   - approved application

3. **APPLICATIONS**
   - assess application
   - valid application

Omit any processing required to handle trivial rejects (i.e., where no work needs to be undone); show the possibility of trivial rejects with a data flow labelled reject which has no destination indicated.

Guidelines for Drawing DFDs

- data flows can diverge when duplicate “packets” of data are sent to different parts of the system

- **APPLICATION**
  - register application
  - reject application

- **APPLICATION**
  - validate application
  - reject application
  - approved application

- **APPLICATION**
  - assess application
  - valid application

- **APPLICATION**
  - customer invoice
  - produce customer invoice

- **APPLICATION**
  - sales order
  - valid sales order

- **APPLICATION**
  - generate shipping slip

Process Modelling Using Data Flow Diagrams

Data flow diagrams generally do not show:

- the flow of control (i.e., triggers and controls on processes)
- the particular implementation sequence (only the input and output data flows should determine the processing sequence shown)
- the timing of processing
- DFDs show the set of possible paths through the processing that occurs within a system from the viewpoint of what happens to the data flows

References