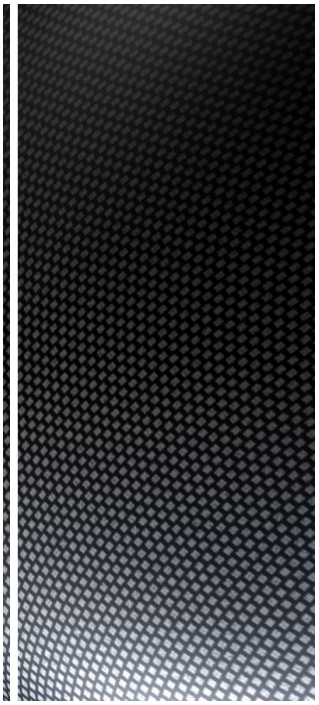
MIS
(Management Information System)

Sharif University of Technology

Session # 6

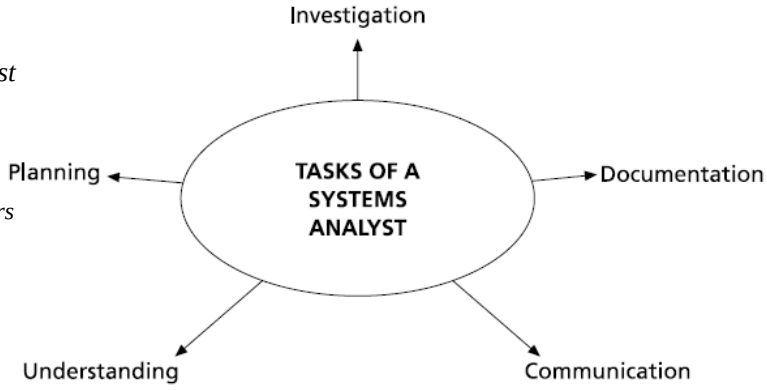


Session schedule

- *Contents*
  - *Systems Analysis and Design*
    - *Planning the approach*
    - *Asking questions and collecting data*
    - *Recording the information*
    - *Interpreting the information collected*
    - *Specifying the requirement*

Information system development

- *System Analysis*
  - *Five areas of a system analyst tasks:*
    - *Investigation*
    - *Communication with customers*
    - *Documentation*
    - *Understanding*
    - *Preparation & planning*



Information system development

- *System Analysis*
  - *System analysis process:*
    - *The PARIS Model*

Analysis can be considered to be a Five-stage process

- *Planning the approach*
- *Asking questions and collecting data*
- *Recording the information*
- *Interpreting the information collected*
- *Specifying the requirement*

Information system development

- *System Analysis*
  - *Asking Questions and Collecting Data*
    - *In order to collect this data and related information, a range of fact-finding methods can be used*
      - *Interviewing,*
      - *Questionnaires,*
      - *Observation ,*
      - *Searching records and document analysis*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Typically, the analyst collects a considerable amount of information during the investigation phase, which may include*
      - *Interview reports,*
      - *Observation records,*
      - *Sample documents,*
      - *Completed questionnaires and*
      - *Lists of problems and requirements.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Structured analysis and design views information systems principally in two ways:*
      - *Data – the information that the system records*
      - *Processing – what the system does with this data.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *The techniques of*
      - *Data flow diagramming and*
      - *Entity modeling,*

    with a data dictionary, are used to record information about the current system.
    - *A requirements catalogue is used to record new requirements.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Data Dictionaries and CASE Tools*

    A data dictionary is used to record all those pieces of information about a system (textual or numeric) that cannot be recorded on diagrams.

    In SSADM, the importance of the data dictionary is recognized in the concepts of
    - *Data flow models (DFMs) and*
    - *Logical data models (LDMs),*

    in which diagrams and their associated background documentation are considered as a whole.

Information system development

- *System Analysis*
  - *Recording the information*
    - *Data Dictionaries and CASE Tools*

    CASE stands for Computer Aided Systems Engineering, and is a term used to describe any software tool designed to make the development of computer systems easier.

    A data dictionary system can therefore be considered a CASE tool, although the data dictionary concept predates that of CASE.

    The term is also used to describe drawing tools specifically designed for the production of the types of diagram used in structured methods.

    CASE tools also incorporate drawing tools and a data dictionary, making it possible to check sets of diagrams for consistency and to see the results of changes.

Information system development

- *System Analysis*
  - *Recording the information*
    - *Data Flow Diagrams*

Data flow diagrams are the most commonly used way of documenting the processing of current and required systems.

They are a pictorial way of showing the flow of data into, around and out of a system.

A complete set of DFDs provides a compact top-down representation of a system, which makes it easier for users and analysts to envisage the system as a whole.

Information system development

- *System Analysis*
  - *Recording the information*
    - *Data Flow Diagrams- DFD components*

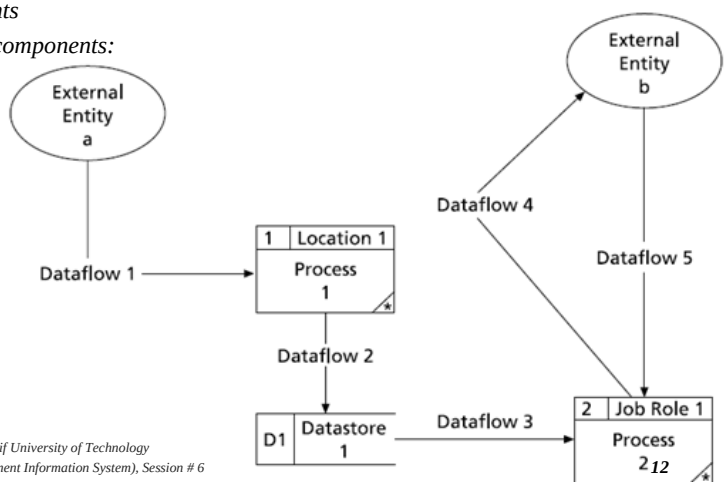
DFDs are constructed using four major components:

External entities,

Data stores,

Processes and

Data flows.



Information system development

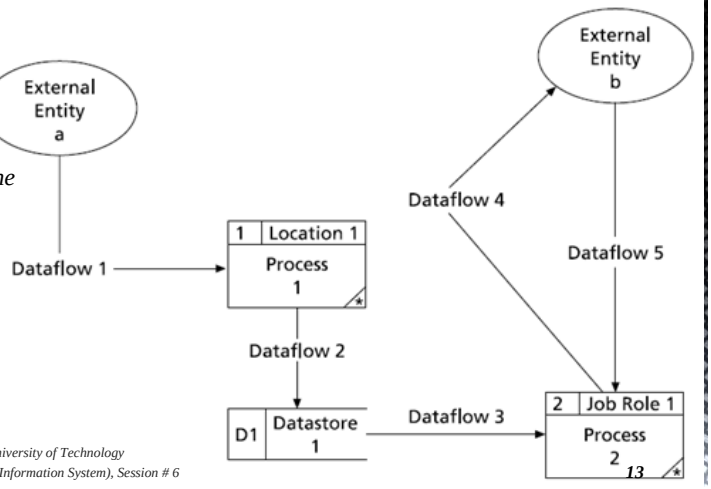
- *System Analysis*
  - *Recording the information*
    - *Data Flow Diagrams- DFD components*

DFDs are constructed using four major components:

External entities,

represent the sources of data that enter the system or the recipients of data that leave the system.

Examples are clerks who enter data into the system or customers who receive letters produced by the system.



Information system development

- *System Analysis*
  - *Recording the information*
    - *Data Flow Diagrams- DFD components*

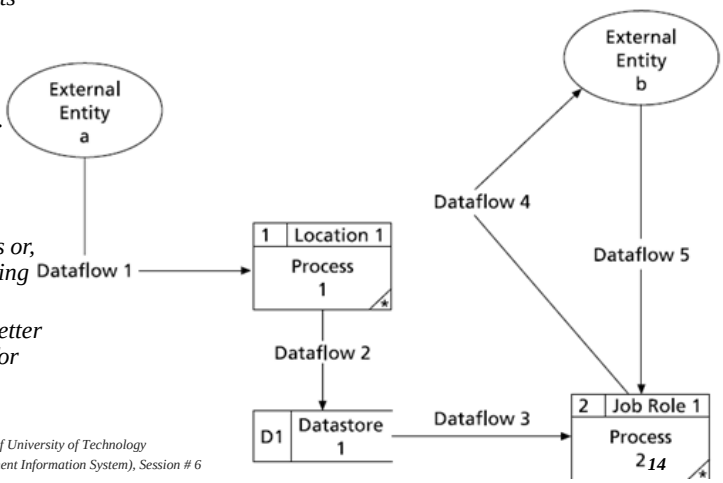
DFDs are constructed using four major components:

Data stores,

represent stores of data within the system.

Examples are computer files or databases or, in a manual system, paper files held in filing cabinets.

Manual data stores are identified by the letter M followed by a number, and identifiers for computerized stores are prefixed by a D.



Information system development

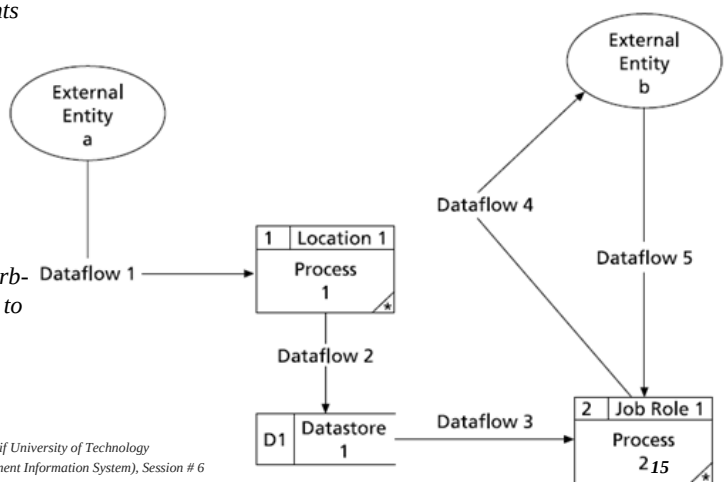
- *System Analysis*
  - *Recording the information*
    - *Data Flow Diagrams- DFD components*

DFDs are constructed using four major components:

Processes,

represent activities in which data is manipulated by being stored or retrieved or transformed in some way.

Some practitioners insist on the use of verb-fronting phrases for process descriptions to ensure that the process is an action performed by the business.



Information system development

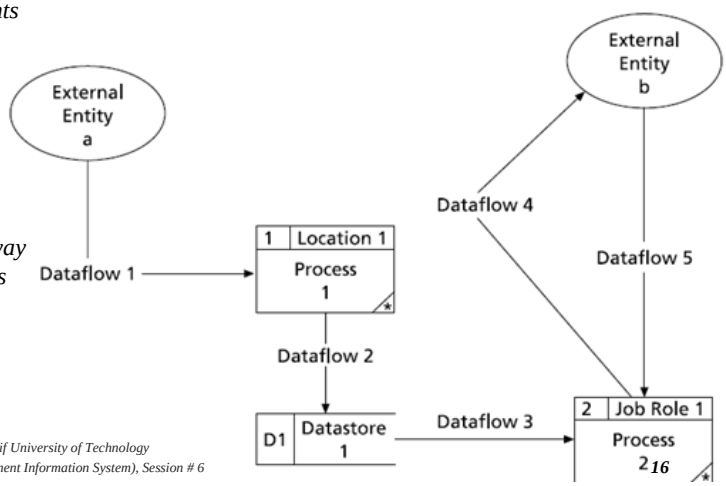
- *System Analysis*
  - *Recording the information*
    - *Data Flow Diagrams- DFD components*

DFDs are constructed using four major components:

Data flows,

represent the movement of data between other components

Data flows are generally shown as one-way only. Data flows between external entities are shown as dotted lines.



Information system development

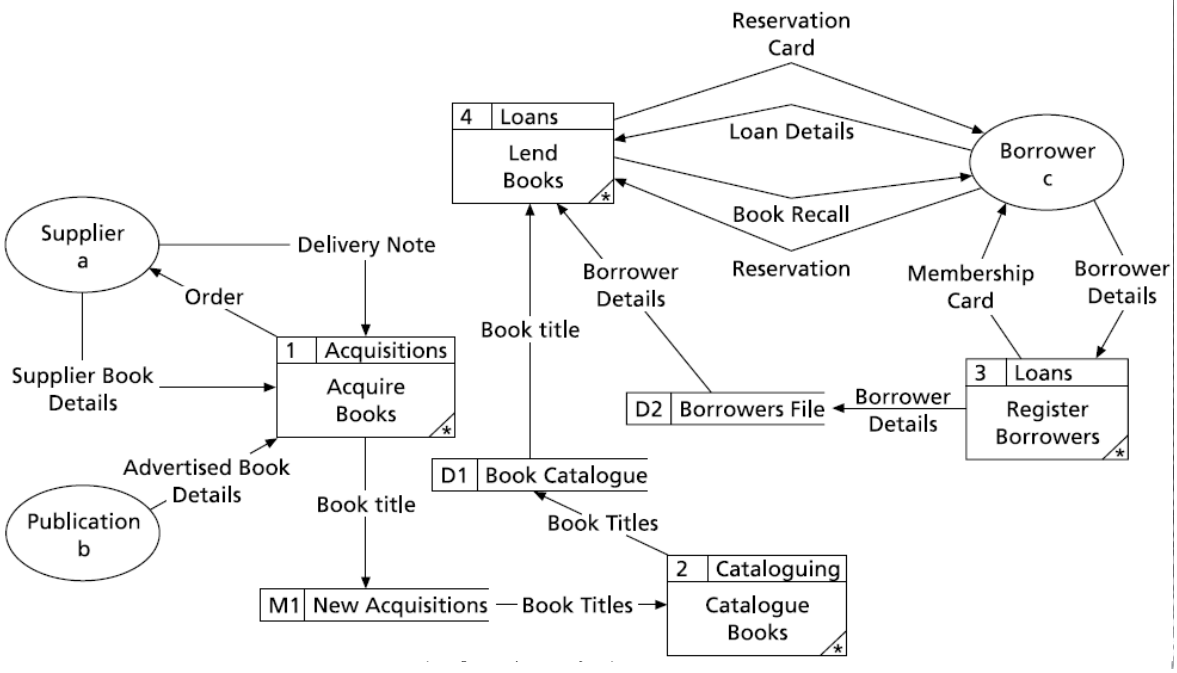
- *System Analysis*
  - *Recording the information*
    - *Data Flow Diagrams*

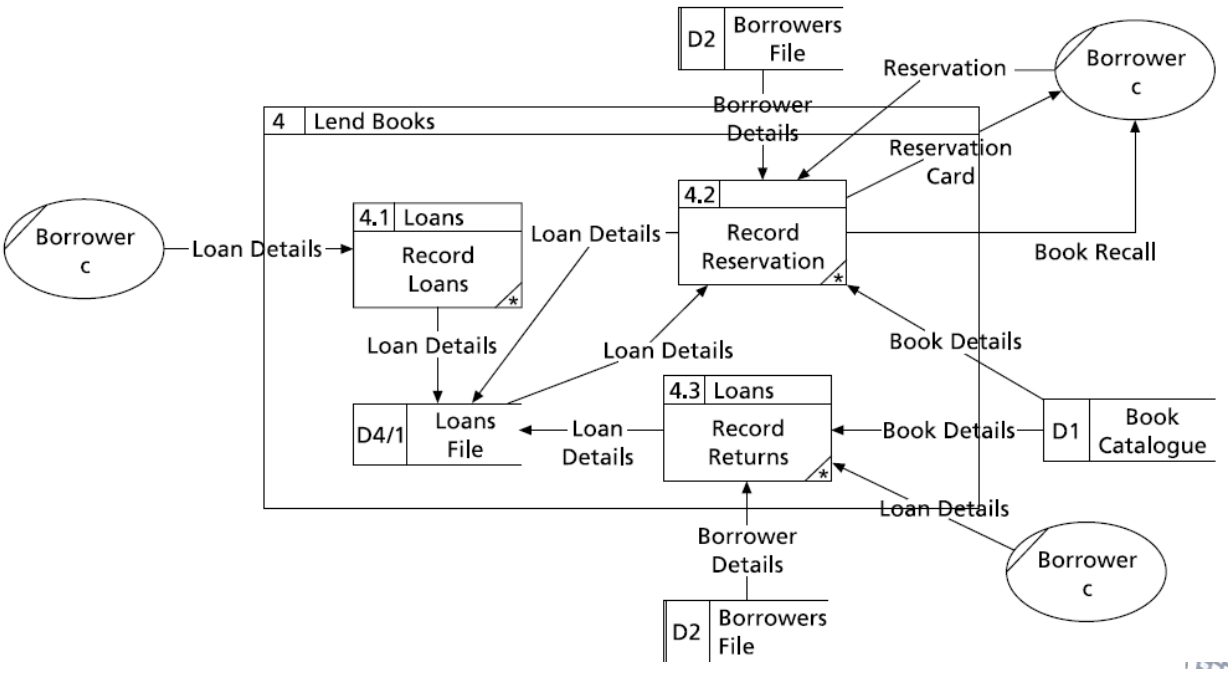
A system is rarely simple enough to be shown on a single DFD, and so a hierarchical set is produced.

This consists of a top-level DFD in which the processes are major system functions described in more detail by one or more associated lower-level DFDs.

The process of breaking a higher-level (parent) DFD into its constituent lower-level (child) DFDs is known as leveling.

As a rule of thumb, however, processes on the lowest level, called elementary processes, should correspond to single events or actions affecting the system, for example cashing a cheque in a banking system.





Information system development

- *System Analysis*
  - *Recording the information*
    - *Data Flow Diagrams*

Diagrams alone are not sufficient to convey an understanding of the system being modeled.

It is important that DFD components are described in more detail than can be conveyed by the short names they are given in the diagram.

Data flows consist of a list of data items that must be documented in a data dictionary so that the data items can be related to attributes in the data model.

Information system development

- *System Analysis*
  - *Recording the information*
    - *Data Flow Diagrams*

    only external data flows – data flows into and out of the system – need to be described in detail.

    These input/output (I/O) descriptions contain:

    • *identifier of the object from which the data flows (e.g. the process number);*

    • *identifier of the object to which the data flows;*

    • *data flow name;*

    • *description of the data flow.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Data Flow Diagrams*

    In a similar way only the bottom level – the elementary processes – need to be described, as this is where the detailed processing occurs.

    Elementary process descriptions consist of:

    • *process number;*

    • *process name;*

    • *description of the process – this may be a simple textual description, or may use more rigorous techniques such as structured English or decision tables.*

    External entity descriptions contain:

    • *external entity identifier;*

    • *external entity name;*

    • *description of the external entity.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Starting the Data Flow Diagrams*

    Two areas may cause difficulty when starting out.
    - *One is defining the boundary of the system. What are the parts of the business to be included in the system?*
    - *Once the system boundary has been defined, what are the boundaries of the top-level functional areas included in the system?*
  - *Two subsidiary techniques help to answer these questions:*
    - *Context diagrams and*
    - *Document flow diagrams.*

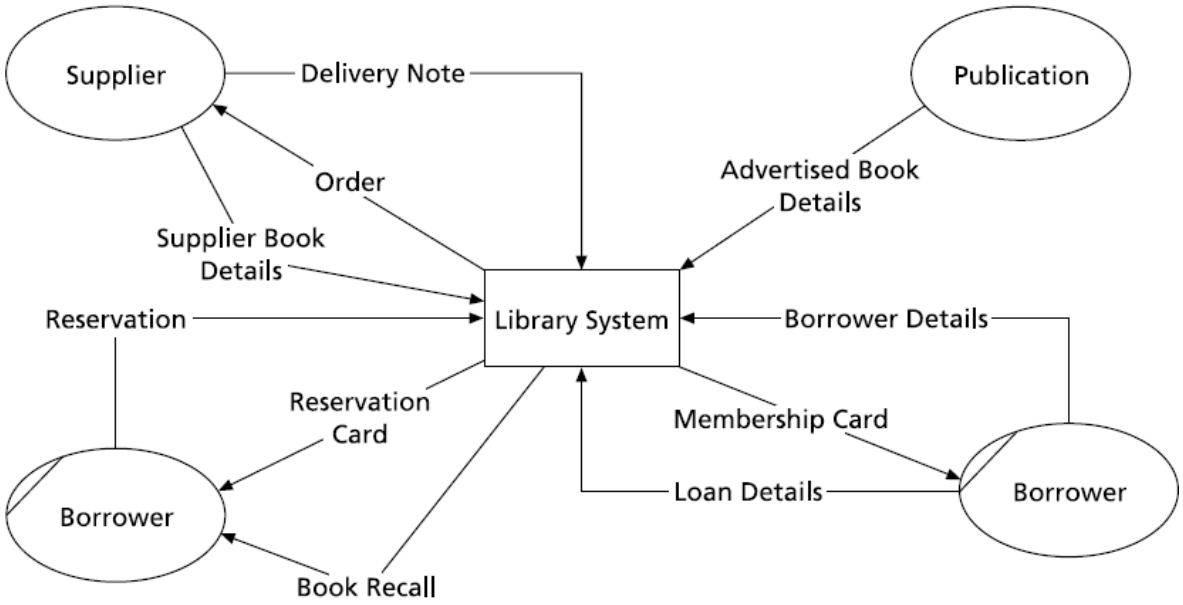
Information system development

- *System Analysis*
  - *Recording the information*
    - *Context Diagrams*

    A context diagram is similar to a top-level DFD but with the whole system shown as a 'black box'.

    In other words, external entities and data flows into and out of the system are drawn but no processes or data stores are shown.

    Context diagrams are used early in a project as a means of describing and agreeing the scope or boundary of the system to be developed.



Information system development

- *System Analysis*
  - *Recording the information*
    - *Document flow Diagrams*

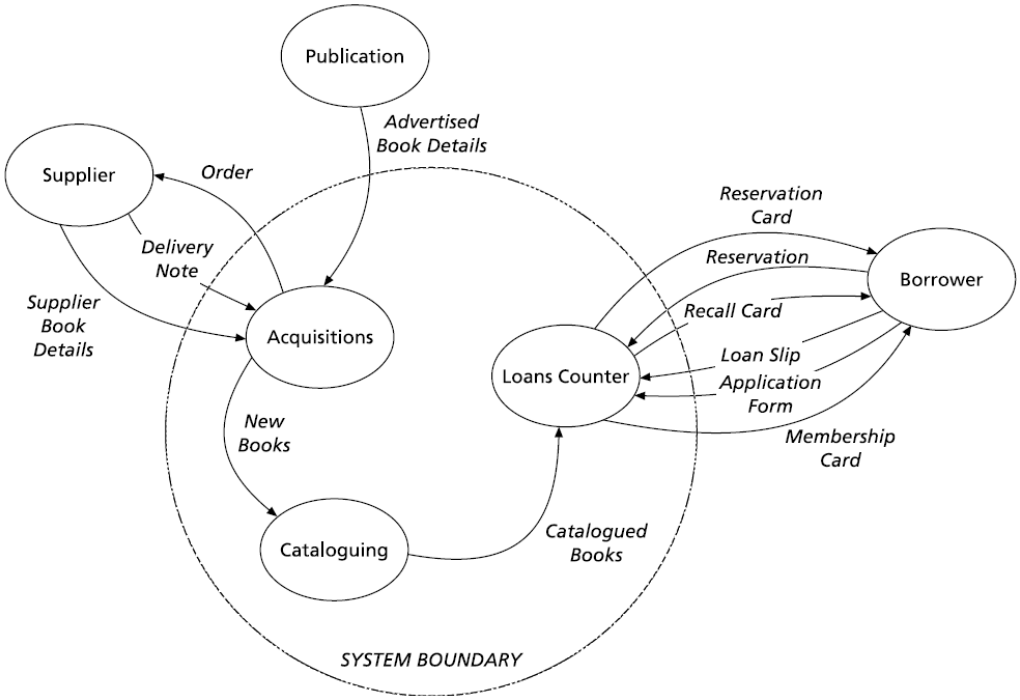
Document flow diagrams may be used as a preliminary to producing DFDs for the current system in the early stages of a project.

They are used to show how documents move round in a manual system.

By drawing boundaries around parts of the diagram, different functional areas of the system can be distinguished.

Areas of the diagram outside the boundaries are external to the system and will appear as external entities and external data flows on the current system DFDs.

The bounded areas will appear as processes on the level 1 DFD.



27

Information system development

- *System Analysis*
  - *Recording the information*
    - *Data flow Diagrams*

Top-level processes are usually fairly easy to identify as they often correspond to departments in the organization.

When the top-level processes have been identified they should be looked at in more detail to see if they need to be broken down further.

The aim is to break the processing down until the bottom-level processes each handle a single event such as the return of a book.

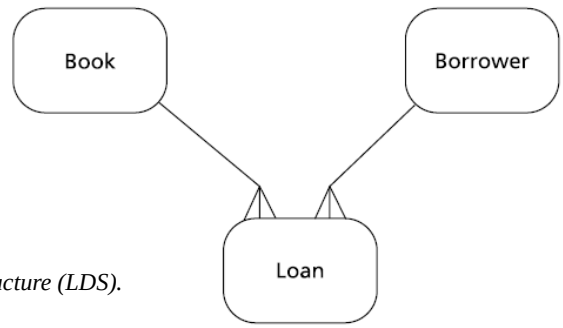
Information system development

- *System Analysis*
  - *Recording the information*
    - *Data flow Diagrams*
    - *Entity Models*
    - *An entity model represents the network of relationships between classes of things that need to have data recorded about them in the system.*
    - *The term entity type or entity is used to describe a 'class of things'.*
    - *entities represent not only physical objects but also conceptual objects, such as 'bank account', or events, such as 'transaction'.*

Information system development

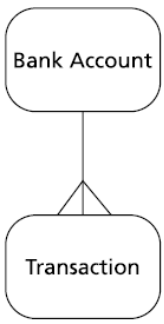
- *System Analysis*
  - *Recording the information*
    - *Data flow Diagrams*
    - *Entity Models*

The entity model in SSADM is called logical data structure (LDS).



LDSs are simpler than DFDs in that they have only two major components – entities and the relationships between them.

An entity is usually represented as a rectangle containing its name, written as a singular noun.



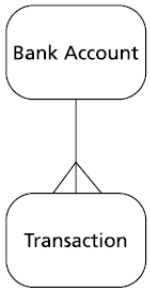
Relationships are shown as lines linking entities. Relationships can be traversed in both directions, and so each end of a relationship is named in order to describe it from the point of view of the entity at that end.

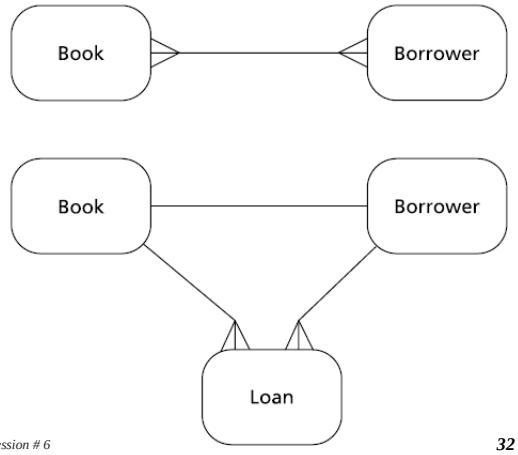
Information system development

- *System Analysis*
  - *Recording the information*
    - *Data flow Diagrams*
    - *Entity Models*
    - *Classification of relationships is known as the degree of the relationship*
    - *Most relationships are between one master entity and many detail entities. <<one-to many relationship>>*
    - *Relationships may also be <<many-to-many >> or <<one-to-one>>.*
    - *One-to-one relationships are uncommon, as it is usually found that two entities that are linked in this way can be combined to give a single entity.*
    - *Many-to-many relationships are more common, but it is usual to resolve them by introducing a new 'link' entity that is a detail of the two original entities*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Data flow Diagrams*
    - *Entity Models*



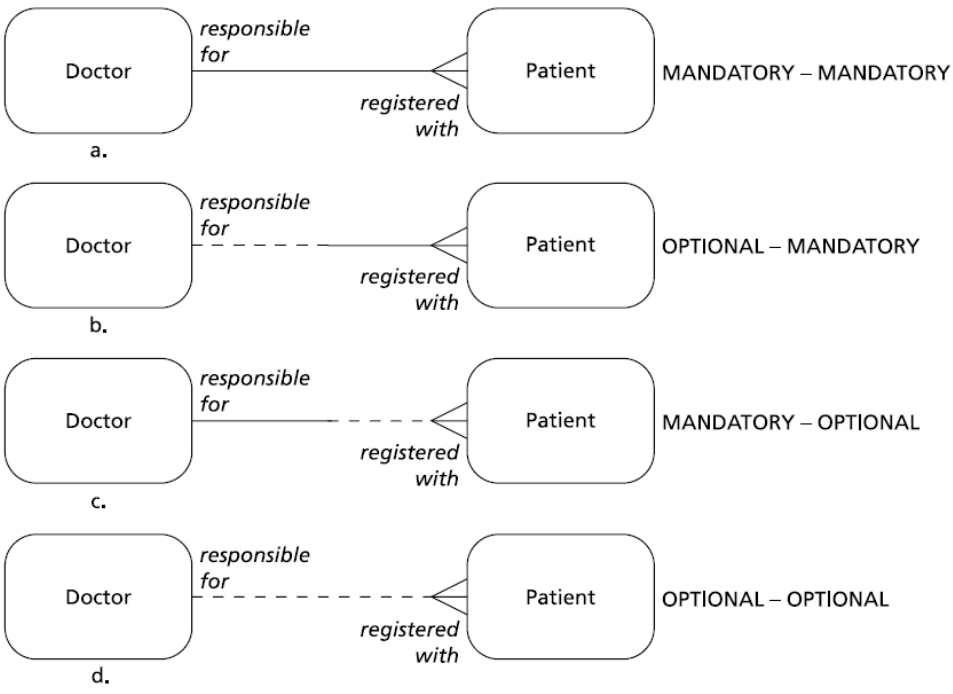


Information system development

- *System Analysis*
  - *Recording the information*
    - *Data flow Diagrams*
    - *Entity Models*
    - *Relationships are also classified in terms of their optionality.*
    - *Relationships can be described as exclusive.*

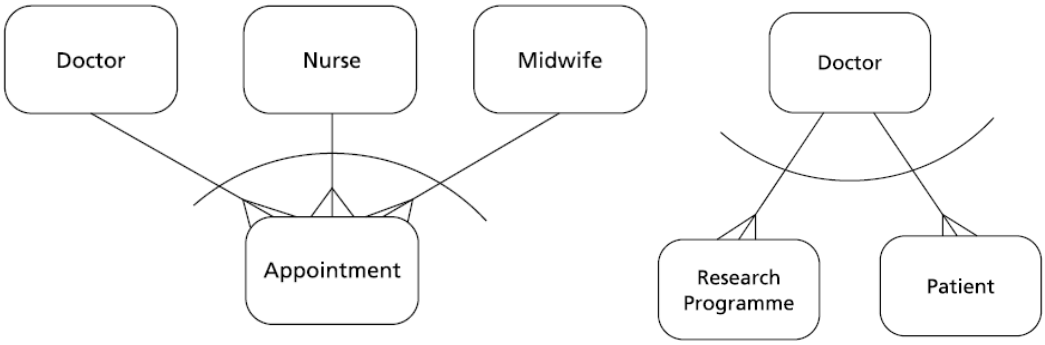
One type of exclusivity occurs if a detail entity has two (or more) masters and an occurrence of the detail may be linked to only one of the masters but not both.

The other is the converse situation where a master may be linked to only one of two or more sets of details. Exclusive relationships are shown by drawing an exclusion arc to connect them



Information system development

- *System Analysis*
  - *Recording the information*
    - *Data flow Diagrams*
    - *Entity Models*



Information system development

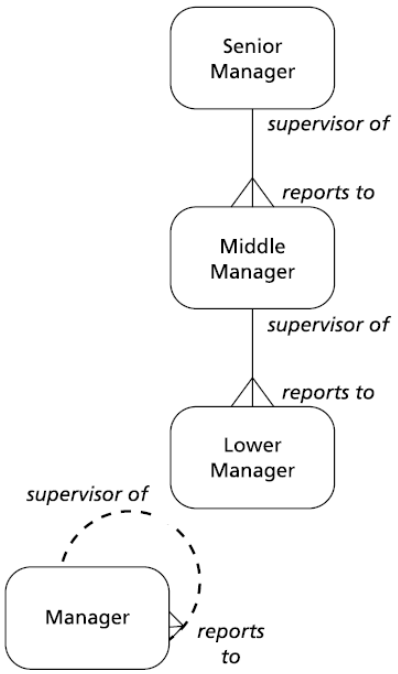
- *System Analysis*
  - *Recording the information*
    - *Data flow Diagrams*
    - *Entity Models*

    - *<<Recursive relationships>>: In other words, individual occurrences of entities can be related to other occurrences of that entity.*

  There are two ways in which this can happen.

  The first is where there is a one-to-many or hierarchical relationship between entity occurrences.

  or by a single entity called manager, which has a recursive relationship with itself.



Information system development

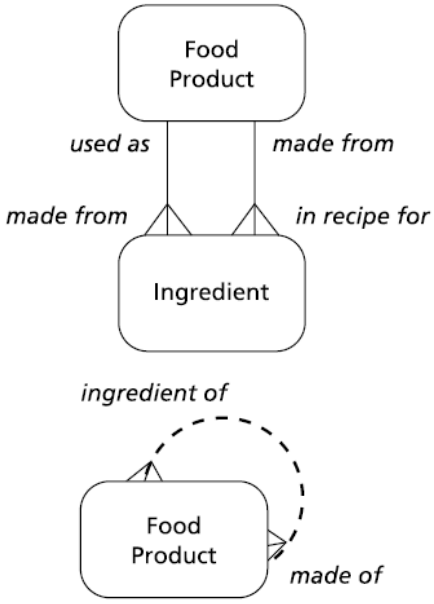
- *System Analysis*
  - *Recording the information*
    - *Data flow Diagrams*
    - *Entity Models*

    - *<<Recursive relationships>>: The second way in which an entity can be related to itself is where there is a many-to-many relationship between occurrences, indicating a network structure.*

    There are two ways in which this can happen.

    One way in which the structure can be shown is as a single entity linked to itself by a many-to-many relationship.

    breaking the many-to-many relationship into two one-to-many relationships and creating a new entity that acts as a link between different occurrences of the original entity.



Information system development

- *System Analysis*
  - *Recording the information*
    - *Data flow Diagrams*
    - *Entity Models-The Logical Data Model*

    - *A logical data model (LDM) consists of an LDS diagram and a set of entity descriptions and relationship descriptions, which give more detail about the diagram components.*

    - *Relationship are usually documented in both directions. In other words, they are described from the point of view of each of the two entities that make up the relationship.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Data flow Diagrams*
    - *Entity Models-The Logical Data Model*
    - *For each relationship 'half', the following information should be recorded:*
      - *First entity name (the entity at this end of the relationship);*
      - *Second entity name (the entity at the other end);*
      - *Relationship name or phrase shown on the LDS;*
      - *Description of the relationship (in business terms);*
      - *Degree of the relationship (one to many, many to one, etc.);*
      - *Cardinality of the relationship (the number of second entities expected to be*
      - *Linked to each first entity – this may be an average or, better, a distribution);*
      - *Optionality of the relationship;*
      - *List of users and their access rights to the relationship (update, read, etc.).*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Data flow Diagrams*
    - *Entity Models-The Logical Data Model*
    - *Entity descriptions should contain at least the following information:*
      - *Entity name;*
      - *Alternative names (synonyms);*
      - *Description of the entity;*
      - *The owner (this is the user to whom the data in the entity belongs);*
      - *List of users and their access rights to the entity (update, read, etc.);*
      - *Expected number of occurrences of the entity and growth rates;*
      - *Rules for archiving and deleting entity occurrences.*

Information system development

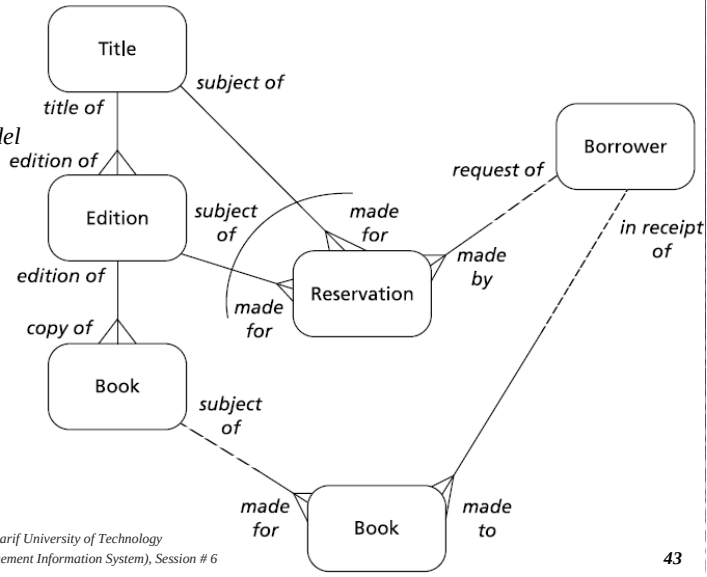
- *System Analysis*
  - *Recording the information*
    - *Data flow Diagrams*
    - *Entity Models-The Logical Data Model*
      - *One of the things we also need to record about an entity is the list of attributes it contains.*
      - *An attribute, or data item, is a piece of information that the system needs to record about an entity.*
      - *Attributes may be held by an entity purely as information, or they may play a role in relationships between entities, in which case they are known as key attributes or keys.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Data flow Diagrams*
    - *Entity Models-The Logical Data Model*
      - *Keys are principally of two types: prime keys and foreign keys.*
        - *Prime keys are used to identify different occurrences of the same entity.*
        - *Foreign keys are attributes that are also present as prime keys on other entities.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Data flow Diagrams*
    - *Entity Models-The Logical Data Model*



Information system development

- *System Analysis*
  - *Recording the information*
    - *Data catalogue*
    - *The data catalogue is a list of all the data items or attributes that have been identified as being required in the system.*
    - *Attributes are the individual items of data that are used to describe entities in the logical data model and which travel along data flows in DFMs, where they are listed on the I/O descriptions.*
    - *The data catalogue is in fact a subset of the data dictionary and is concerned with individual data items and the values they may take*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Data catalogue*
    - *The information that should be recorded about attributes includes:*
      - *Attribute name;*
      - *Alternative names (synonyms);*
      - *Description of the attribute;*
      - *Attribute location (entity or data flow);*
      - *Relationships to other attributes;*
      - *Format (including units and length);*
      - *Values (or ranges of values) the attribute is allowed to have;*
      - *Rules for deriving the value of attribute occurrences;*
      - *Optionality of the attribute;*
      - *The owner, i.e. the user to whom the data in the attribute belongs;*
      - *List of users and their access rights to the attribute (update, read, etc.).*