ISD1
Introduction to Data flow diagrams Lecture 1 and 2

Martina A. Doolan
M.A.Doolan@herts.ac.uk

Questions to Answer?

- What are Dataflow diagrams?
- What is their purpose?
- How do I draw them?
- The notation – syntax (components)?
- The semantics (meaning)?
- Whats allowed? Legal/illegal
- Levelling?
- Numbering/Labelling?

What are Data flow diagrams?

"A data flow diagram ... the picture worth a thousand words." (Tom De Marco)

- So..

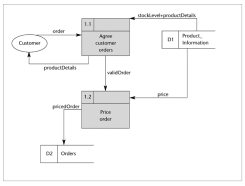
A picture of the FLOW OF DATA through a system of any kind showing the EXTERNAL SOURCES and DESTINATIONS of data, the PROCESSES which transform the data and the places where the data is stored (DATASTORES)

Data flow diagrams?

OR…
A model which describes the behaviour of a complex system in terms of smaller, more manageable chunks, each of which can be "held in one's head".

An illustration – A picture!

Figure 4.2 Fragment of a data flow diagram



The Data flow diagram…

…is a diagram which summarises:

- Data movement
- Data sources
- Data recipients
- Data Storage
- Processes which act on data

Why use them?

- Easy to follow
- Show system boundary
- Enables the system to be shown at several different levels
- Working document to be revised and updated
- A means of communication
- Basis for program specification

7

What is their purpose?

- identify the *system boundary* and *external entities*
- models the *data or information flows* into and out of the system
- chart all possible data routes through the system
- model what the system does
- divide the system into 'brain-sized chunks' – the data flow diagram is the major partitioning tool.

8

Basic elements of a DFD?

- Data flows
- Processes
- Data stores
- External entities
- System boundary

9

The data flow:

- models the connections or interfaces between components of the dfd
- models the *data or information flows* into and out of the system
- is represented as an arrow with a unique label
- represents a group or packet of data items

10

A Process

- models something happening to the data
- is shown as a rectangle
- transforms incoming data flows into outgoing data flows- it processes the data
- will typically have one or more data inputs and produce one or more data outputs

11

A Process...

- will have a unique number and a brief description of their function, e.g. Agree customer orders, Price order.
- May be labelled with its location such as 'Personnel , 'Accounts' or sometimes with a description of who does the work e.g. 'Receptionist'.

12

A Data Store

- models represents permanent data used by the system
- processes can enter data into a data store or retrieve data from a data store
- each data store has a unique name and reference number

13

An External Entity

- represents something outside the system boundary, often a person or organization
- sometimes referred to as a *source* or a *sink*
- either supplies data to the system (source) or receives output form the system (sink)
- shown as an ellipse
- System boundary is shown as a dashed line

14

Data flow diagram notation…

- several different notations for dfds
- no definitive standard
- only the shape of the symbols vary, not the underlying logic
- notation used on this course follows SSADM standards – Britton's book

15

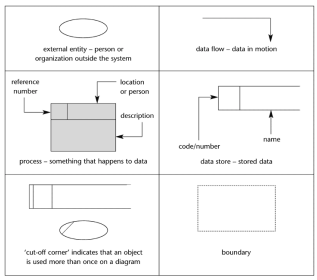
Our approach on this module…

- We will use the notation in Britton & Doakes book

Which is…..

16

Notation



17

Syntax

- Data flow diagrams provide us with a *language*, which, like any other language, has a *syntax* (a set of "grammar rules"). The syntax of DFDs tells us how the symbols can legally be fitted together
- The syntax tells us which combinations of symbols are "legal": any combinations not allowed by the syntax rules are "illegal".

18

Semantics - meaning

- If it is to be of practical use, a language also needs to have a *semantics* (a set of rules for interpreting what legal combinations of symbols are intended to *mean*.)

19

So, when we construct a data flow diagram, we have to :

- make sure it is syntactically correct: that is, we must link the symbols together in "legal ways":
- If a DFD is not syntactically correct it has no "meaning" in terms of the language of data flow;

20

So, when we construct a data flow diagram, we have to :

- make sure it provides "meaningful" information about the particular system we are trying to represent:
  - It is essential that we define the label attached to each data flow, and the type of data contained in each data store, in a data dictionary.
  - When we *choose* labels for flows, processes and so on, we try to make sure that they are as meaningful to the reader as possible: that is, we call our flows by names like *Order* and *Statement*, not names like "Y"!

21

Legal combinations

Process sends data to Process
Process receives data from Process
External Entity sends data to Process
External Entity receives data from Process
Process sends data to datastore
Process receives data from datastore

- Two-way flows - one from and one to a datastore (though for clarity better not to use this)

22

Illegal combinations

- Entity sends data to Entity
- Entity receives data from Entity
- Entity sends data to datastore
- Entity receives data from datastore
- Datastore sends data to datastore
- Datastore receives data from datastore
- Two-way flows (except one from and one to a datastore (though for clarity better not to use this)

23

Levelling

- The idea of levelling is to allow us to view the system at different levels of detail - to offer a general overview of the system and selective viewing in progressively greater detail
- we use maps in the same way. Maps of different scales allow us to view
- the whole world at one time
- a country in more detail

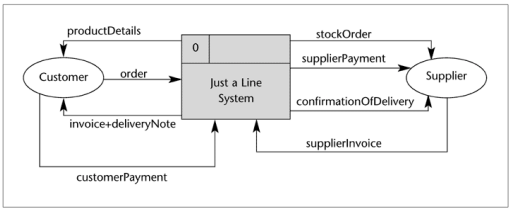
24

Context Level (Level 0)

- The top-level diagram, level 0, is known as the context diagram.
- Models the whole system as a single process box whose sides represent the boundary of the system.
- Identifies all external entities and related input and output flows.
- Defines the boundary of the system and so delineates the domain of study

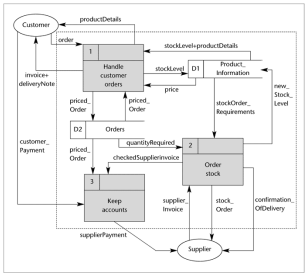
25

An example Level 0



26

Level 1



27

Expanding processes

- The level 1 data flow diagram:
- identifies the major system processes and data flows between them
- expands the single process of the context diagram (level 0)

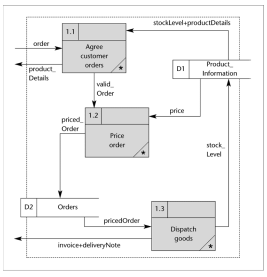
28

Expanding...

- partitions the system into its main constituent parts, each part being an identifiable function or group of functions
- external entities **need** only be shown on the context diagram
  each of the level 1 processes can in turn be expanded or decomposed into a level 2 data flow diagram to reveal more detail

29

Level 2 expansion of Level 1



30

Numbering

- Each dfd is labelled with:
- its level number
- the name of the process it describes
- its stage – current, required, logical or physical
- on the level 1 diagram processes are numbered 1,2,3...

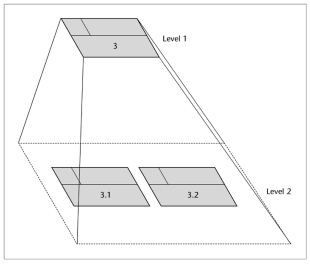
31

Numbering...

- If a process at any level is decomposed, the sub-process numbers are prefixed with the number of the process they decompose. For example, if process 2 at level 1 (see Figure 7) is decomposed, the processes at level 2 will be numbered 2.1, 2.2 etc

32

Figure 4.7 Relationship of a process to its sub-processes – Britton & Doakes



33

Labelling

- This text uses a standardized notation for labelling.
- Labels for data flows are written in lower case;
- where a label consists of more than one word, the words are run together with no spaces but with a capital letter at the start of the second and any subsequent new words e.g. customer name becomes customerName.

34

Labelling....

- Labels for data stores use the same convention except that the first word is capitalized e.g. ProductInformation.
- External entities are labelled in the same way as data stores.
- Process labels should be short sentences describing what the process does, e.g. Price order and Order stock. Normally they contain an active verb.

35

Labelling...

- Data flows should be labelled with nouns representing the data items flowing along them, for example order and customerPayment.
- Data stores labels should indicate the type of data stored in them.
- Each data store has a unique reference number. To distinguish them from processes, data stores are often numbered D1, D2 etc or M1, M2 for manual data stores

36

Labelling.....

- Labels should carry as much meaning as possible, but should be short, to avoid cluttering the data flow diagram.
- Data flows and the contents of the data stores should be documented in a data dictionary (discussed later on this module)

37

Conclusion

- Dataflow diagrams model the flow of data through the system. Used to design the system at different stages of development. Mostly used in the requirements engineering stage in the development lifecycle. May also be used to model the system during the feasibility stage and at the design stage.

38

And finally...

- Make sure that understand all the material presented today. Follow the module schedule, do all the reading and exercises. These slides are available on Studynet.
- The tutorial next week will give you an opportunity to practice doing a DFD using a case study. Come along to the tutorial prepared, any questions feel free to ask on Studynet or ask your group tutor during your next tutorial.

39

References

- Britton, Carol, Doake Jill, Software Systems Development : A gentle introduction 3rd ed. Mc Graw Hill, ISBN 0-07-709974-5

40