What do we do next ....

- Once your functional specification is approved i.e. the user believes that you understand what they want ....

WHAT DO YOU DO?
- ignore it ... and start designing from scratch
- use it as a basis for design ... making changes as required (after approval)

How do you use it?

- The ERD and DFDs become the basis for your database design
- The DFDs become the basis for your interface design

When gathering the data to be represented in the DATA DICTIONARY, for the ERD of 'Schedule Conference Session' we started with the following:

PAPER = Paper # +
Allocated-stream # +
Speaker-person # +

PERSON = Person # +
Contact-details +
{Attendance-day}

STREAM = Stream # +
Stream-name

ERD: Schedule Conference Session

Function 1: Level 1 Schedule Conference Session

1.1 Check request validity
1.2 Display available sessions
1.3 Select best session
We then use the ER model and the DFD to confirm and gather further information about the data and the relationships:

**PAPER** =

- **Paper #:**
- **(Final-allocated-stream #:** (see 1 and 2) Note: This is not the same)
- **Day #:** (see 2) data element as the
- **Start-time #:** (see 2) stream that the author(s)
- **Speaker-person #:** (see 3) initially nominated

From the ER model we are interested in:

1. What session a paper is allocated to, if the allocation has taken place,
   - So we add details about the session:
     - Stream no already exists so we just need to add Day and Start-time
2. What stream a paper is allocated to:
   - So we need stream information
     - (already there)
3. Who is the presenter (speaker) for the paper:
   - So we need speaker information
     - (already there)

From the DFD we are interested in:

1. Providing the paper #: to Process 1.1 – this is available
2. Providing the Paper-allocation-details (Stream, Day, Start-time) to Process 1.2 – this is available
   - (because there is no data dictionary available we are guessing that this is what it might be from the name)

Why do we need Speaker-person #: if it is not being used by this function???? (or was it part of the Paper-allocation-details dataflow) **** Note the importance of the Data Dictionary

Carrying out this process helps us identify that there may be a flaw in the DFD

The reason we do need speaker information is so that we do not allocate the paper to a time slot that the speaker is already scheduled for in a different stream.

So we need to add the appropriate data flows to the DFD.

We carry out this process for each of the entities in our Entity Relationship diagram ensuring that all functional requirements are met:

Example:

**PERSON** =

- Person #: +
- Contact-details +
- {Attendance-day} +
- {Speaker-paper-no} +
- {Manager-session-no} (Stream#, Day, Start-time)

For ease of processing I have decide to create a key for Session called Session #, even though a combination of Stream #, Day and Start-time would uniquely identify a session.

We then normalise each entity:

**PERSON** (Person #, Contact-details, (Attendance-day),
- (Speaker-paper-no), (Manager-session-no))

The normalisation will be done during the seminar......

The above data represents:

A person at a conference may attend many days
A person at a conference could be the speaker for many papers
A person at a conference could be the manager for many sessions

When normalising the ( ) brackets represent repeating groups (many) ... this is a different convention to that used in the Data Dictionary

**EACH OF THE NORMALISED ENTITIES REPRESENTS A TABLE IN YOUR DATABASE**

Using DFDs to develop your interfaces ...

This data flow would become an input on a screen
This data flow would become an output on a screen or a report
This would be an access to the database
This would be an access to the database
This would be an access to the database
This would be an access to the database

Interface development will be done during the seminar......

References


Resources provided for Assignments 1 and 2 – See Seminar section of the IMS2000 website