

1.3.3.9 Scalability & t.c.o

- Take growth into consideration
- Scalability
 - Adapt over time
 - Need to research ~~the~~ current & future growth of inputs processes & outputs
- Transaction volume greatly increases cost
- Take notes of hidden costs

1.3.3.10 Info gathering

- Fact finding
 - Sys requirements
 - Scalability
 - Requirements modelling
 - Work as a JAD group or alone
 - Interviews
 - review docs
 - Observing
 - Preparing questionnaire
 - Surveys
 - Sampling
 - Research
- 1 Determine info needed
 - Ask questions about current sys
 - Propose new sys questions
 - Questions:
 - What inputs
 - Response time
 - Concurrent users
 - Whats needed
 - time
 - Business func.
 - Any older sys need to interact with
 - budget
 - human resources
 - Develop plan to get facts
 - fact finding techniques
 - document results
 - Present to management

1.3.3.11 who what when where & how & why

- Ask who why, what why etc
- Best to understand both current way & right way

1.3.3.12 Other fact-finding methods

- Review docs
 - Get copies of forms & docs being used.
- Observing
 - Compare what's being said vs done
 - Draw up lists of tasks to be observed
 - Hawthorne effect
 - Employees being observed are more productive ✓
- Surveys
 - Heading
 - Title
 - purpose desc.
 - Contact
 - Deadline
 - Should they be anonymous
 - Don't use leading questions
 - Consider anonymity
 - 3 types of questions
 - Closed ended
 - Open ended
 - Range of responses
 - Test on small group
- Sampling
 - Examples of actual docs/data collected to aid in study.
 - eg - reports
 - error logs
 - records
 - requests

- Other forms
- Systematic
 - Every 10th customer
- Stratified
 - 2 customers per province
 - Certain percentage
- Random
 - 18 random customers
- Researching
 - Review docs, journals, tech info, news, trends
- Interviews
 - Establish interview objectives
 - Discuss persons ideas/opinions
 - Construct questions
 - Same per department/team so comparisons can be made
- Prepare
 - Informed of date & time
 - Employees provided with questions that require detailed info.
 - Have samples of docs you may have questions about
- Carry out interview
 - Determine who to interview
 - Intro oneself
 - Describe project
 - Explain your goal for new sys
- Document
 - Take notes
 - Send memo thanking participant
 - Include meeting summary.
- Eval the interview
 - Establish Biases
 - Do they actually know what they

are talking about.

1.3.3.14 Documentation

- Keep accurate interview records
- Managing info is the expectation of good analysts
- Document ASAP
- Use simple rec methods
- Organise docs so related docs easily found
- Use Case tools anytime.

1.3.3.15 Data, process & object modelling

- At end of req. modelling process
 - Should understand business process & sys requirements.
- Model logical design is next step
 - Structured analysis approach
 - Views system
 - data
 - processes
 - how processes act on data
- Object modelling
 - Looks @ sys in terms of objects
 - Contain data & processes

1.3.4 Data & process modelling

View sys as:

- inputs
- outputs
- processes that act on data
- Sys Analysis phase
 - SDLC, data & process models developed
 - Graphically show how data is transformed

Structured analysis's deliverable

- Logical model of what sys does, not how
 - Also called business model
 - Meet user req.s.
 - Support business function
- 3 main tools
 - data flow diagrams.
 - data dictionary.
 - Process desc.

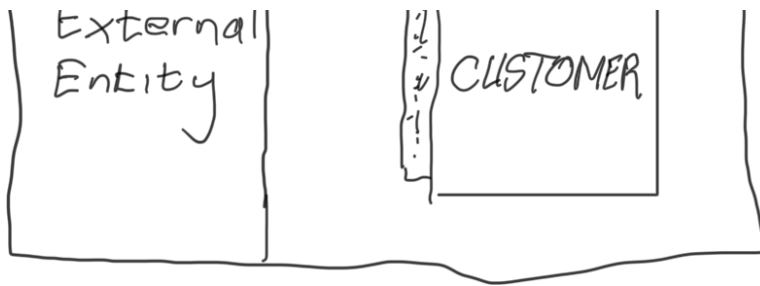
1.3.4.1 Data flow diagrams (DFD)

- Graphical representation of data movement to and from sys and procs acting on data
- Does not show program logic or processing involved
- Only shows what sys does not how.
- By not focusing on how to get sys to do it, focus shifted to finding best design

1.3.4.1 Data flow symbols

- Many styles of DFD symbols
 - lets use Gane & Sarson symbol set





- Sometimes called a black box (the process)
 - Inputs, outputs & processes known
 - Details not shown
- Process usually contains business logic
- Process name always starts with verb & ends with noun

- Data flow symbol
 - Contain noun & adjective
 - eg - PAY INVOICE

- Correct flow
 - data flow entering & exiting process.

- Incorrect flow
 - Process has no input
 - Spontaneous generation process
 - Process has no output
 - Black hole process
 - Wrong input
 - grey hole process

- Data store symbol



- Noun or Plural noun (BIRD FOOD)
 - Collective noun also used on occasion
 - eg EMPLOYEE ROSTER

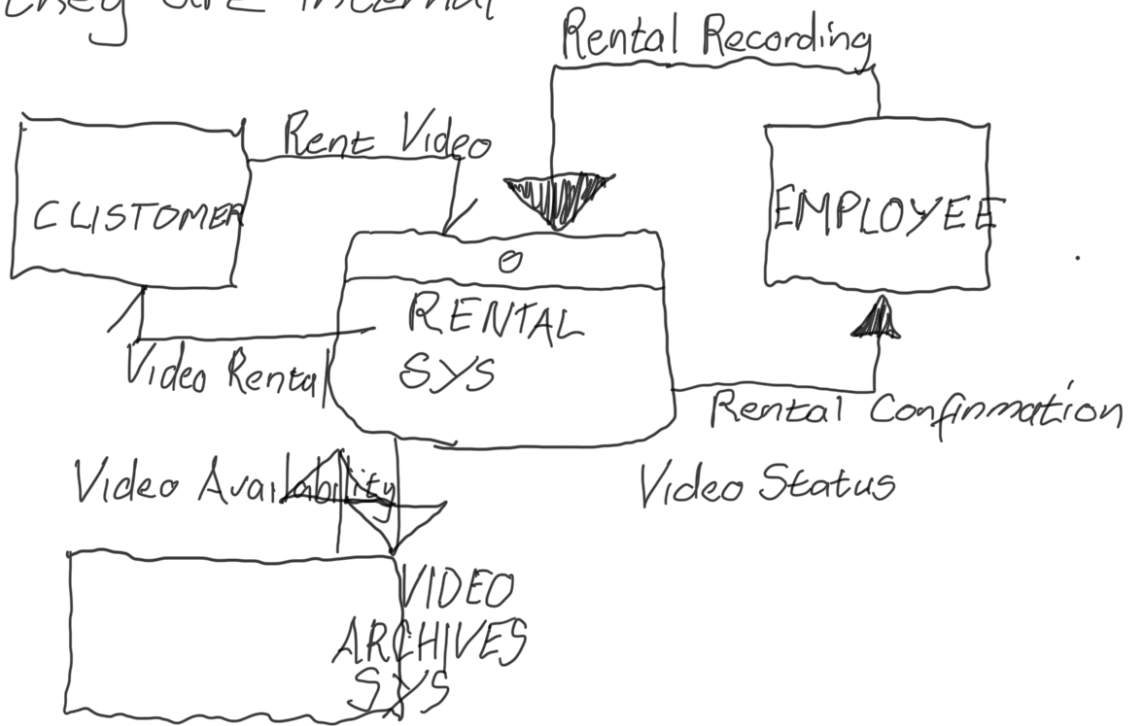
- Must be connected to a process using data flow

- Each data store must have both input & output data stores
- Must also be connected to a process symbol via data flow
- External entity symbol
 - Good indication of where boundaries begin & end
 - Represented by rectangle
 - Also known as terminators
 - Origin & destination of data
 - the source
 - Provides input to system
 - the sink
 - Receives output data
 - Can be both source & sink

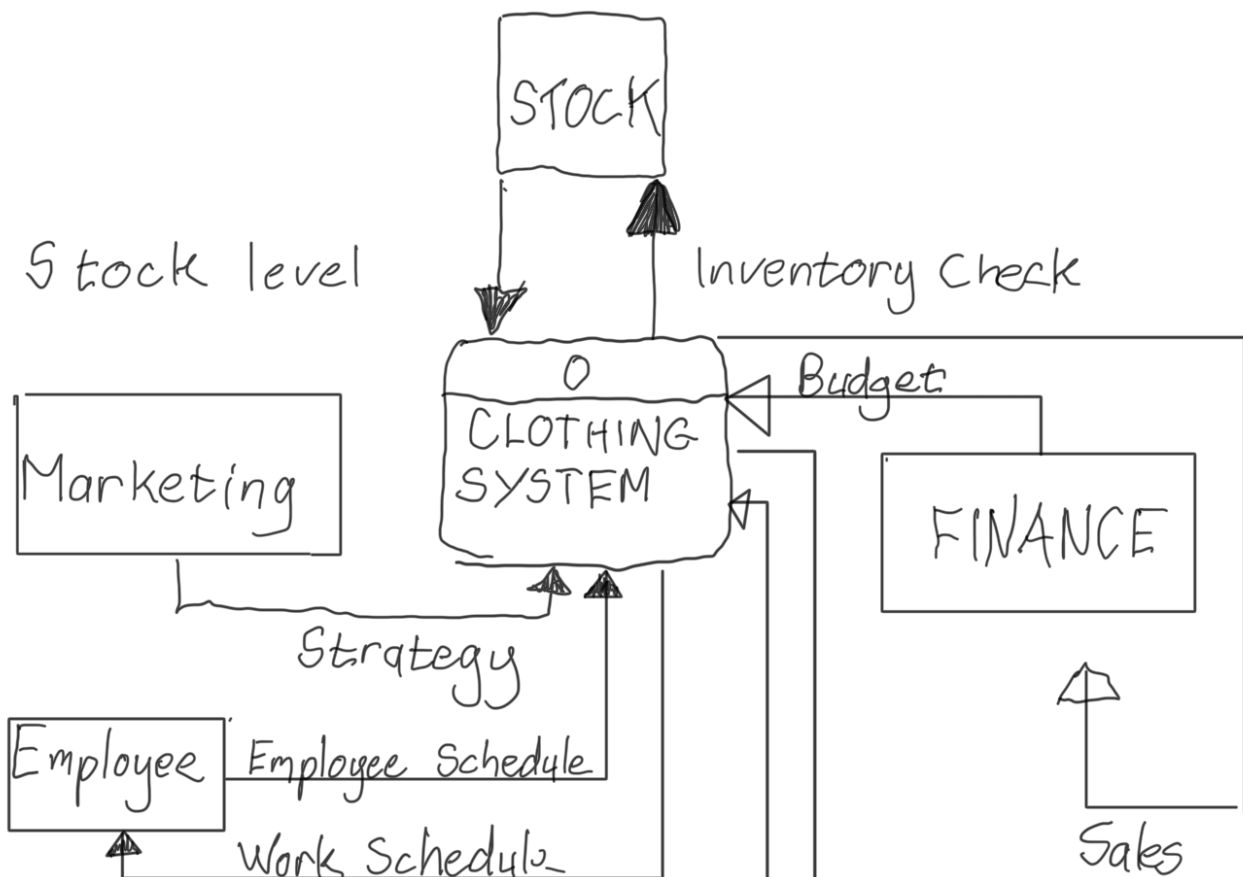
1.3.4.1.2 Context Diagrams

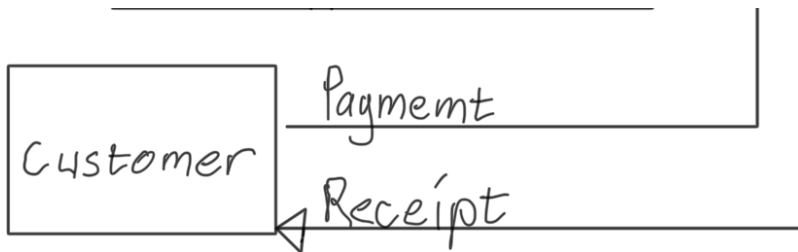
- 1st step when developing a set of DFDs
- draw set of context diagrams,
 - Show sys at top level
 - Where boundaries are and scope of sys
 - 1st step in context diagram
 - Draw single process symbol in centre of page representing sys
 - Called process o
 - 2nd step
 - Place external entities around process o
 - Connect entities to process using data flow
 - Document
 - Data flows
 - direction

- Content
- External entities
- Data stores
- Data stores aren't shown in context diag as they are internal



1.3.4.1.3 Diagram 0



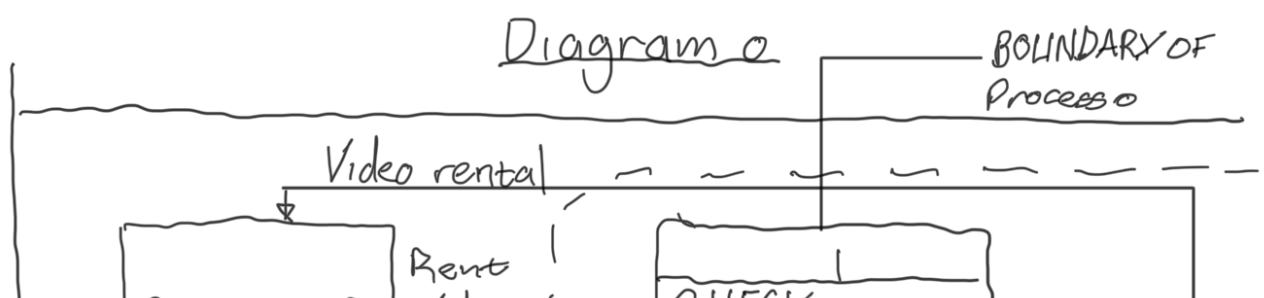
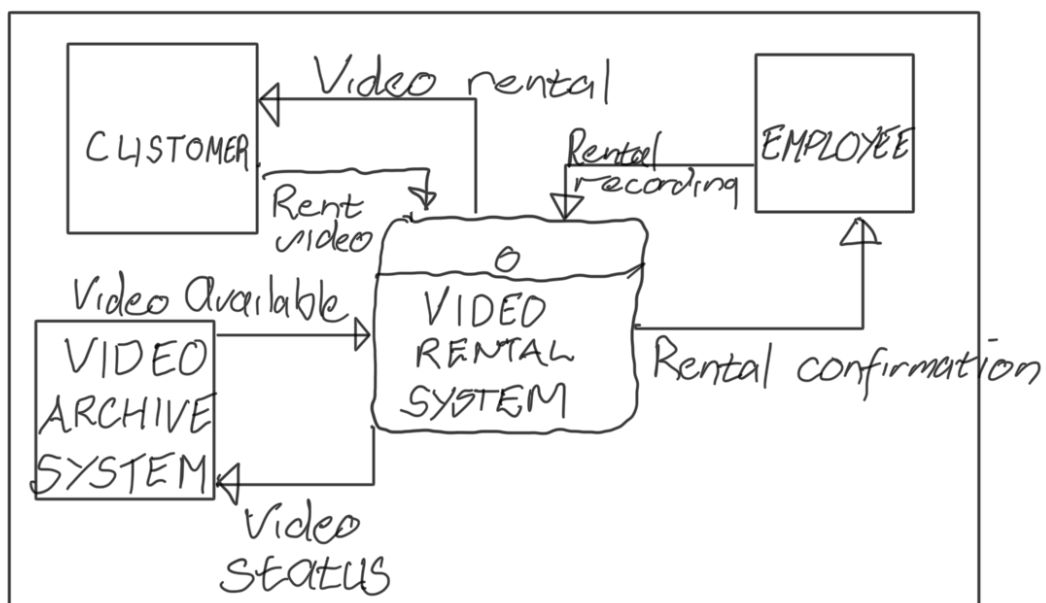


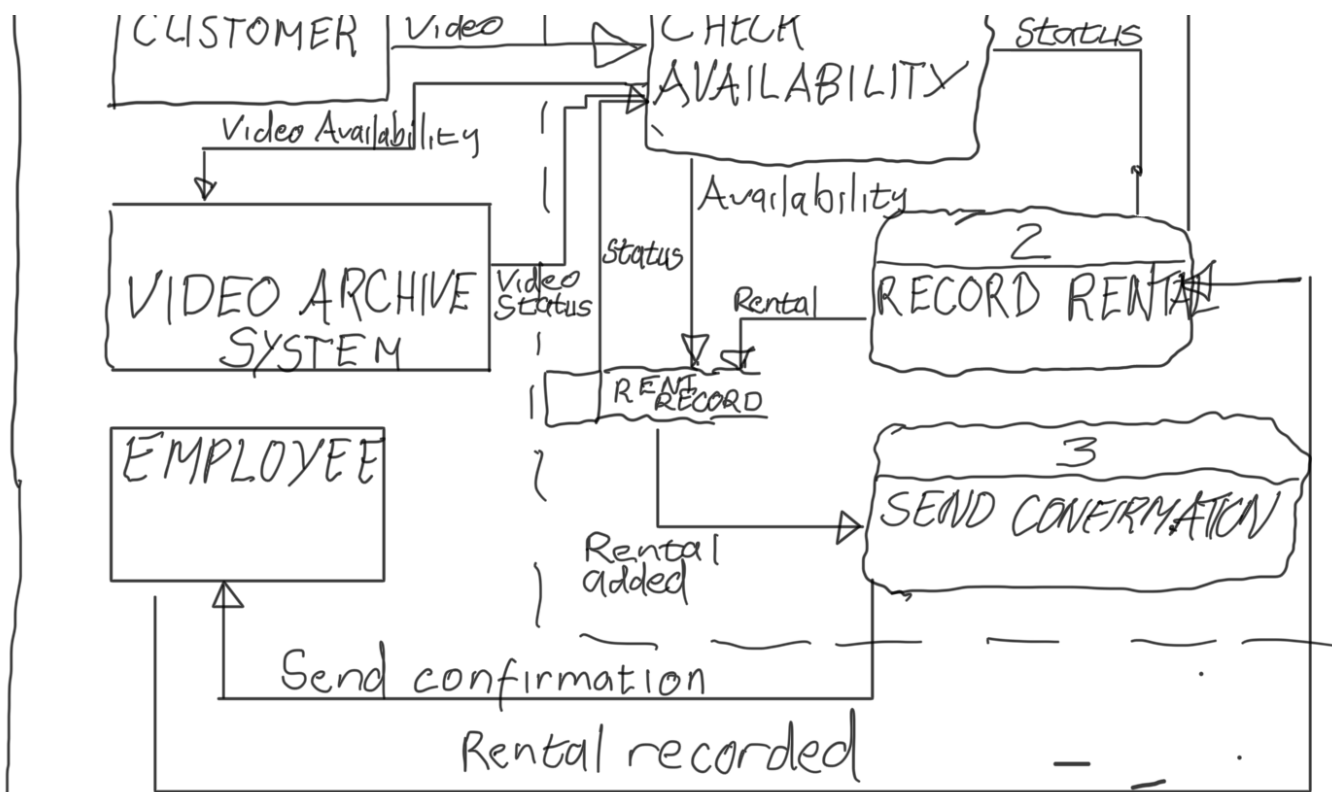
This could be a sys mentioned earlier
 (black box of previous diagram (context diagram))
 gave general sys view

A DFD called diagram 0 (also called exploded, decomposed or partitioned view of process 0)
 Shows context in further detail.

- Shows major processes, data flows and data stores
- Any entities and data flows in context diag repeated
- All connections flowing to and from process 0 must be kept

Context Diagram





- The numbers in the block ID's a series of DFDs for the whole system
eg RECORD RENTAL is 2, a diagram like the above will exist called diagram 2
- The number is merely a tag & not indicative of system process execution order

DFD does not indicate order in which processes are performed, this is found in process desc.

Diverging data flows

- Same data goes to different locations
- Goes from 1 line to 2 or more
- Best to use this approach instead of multiple lines

DFD being expanded

- Called parent diagram
- Diagrams that result from this called child

- lower level diag.

When DFDs are created, processing logic may sometimes be broken down further depending on process complexity

If a process is too complex, it can be broken down into smaller units

- Will be used for code when logical design implemented as physical system

If process can no longer be expanded it's called the functional primitive

- All functional primitive logic is recorded in proc desc which is stored in data dictionary

1.3.4.1.4 Lower Level Diagrams

- Show further processing detail.
 - Diagrams should be leveled & balanced
 - Levelling
 - Create diagrams in series
 - Increases in level of detail shown
 - Can increase until all processes are functional primitives
 - Also called partitioning, decomposing or exploding
 - Balancing
 - Ensures diagrams are consistent with other diagrams
 - Making sure parent input & output flows appear in child
 - All data flows, data stores, external entities and processes must be defined in data dictionary
- Some lower level diags show data stores

- Not visible in parent diag.
- A datastore only showing
 - 2 or more processes use it
 - and that datastore used for > 1 process

1.3.4.1.5 DFD development strategies

- Set of DFDs
 - top down graphical model of sys
 Start with context diagram
 and then create diagram 0 then other nos
- Alt approach is bottoms up
 - All functional primitives are identified
 - Along with data flows, processes, ext. entities
 - Group processes & other symbols and develop lowest level diagram
 - These diagrams are logically grouped together to form next level
 - This continues till diagram 0 is reached
- Diagrams must be approved by users

1.3.4.1.6 DFD Rules & Conventions

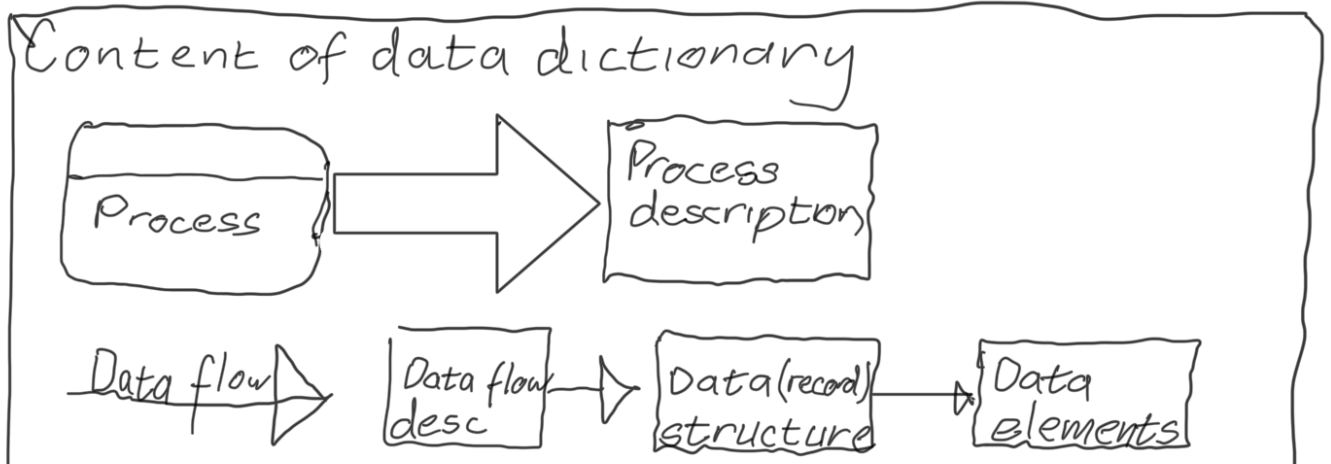
- Each diagram fits on 1 page
- process 0 should be name of system
- Shows entire info sys
- Use unique names
- Don't cross lines
- Limit no of processes
 - ~~Should not be more than 9 symbols~~
- Use unique names in sets of symbols
 A Student in 1 DFD must be the same entity in a parent or child
- Data stores use same naming convention
- Lower level DFD should be verb with

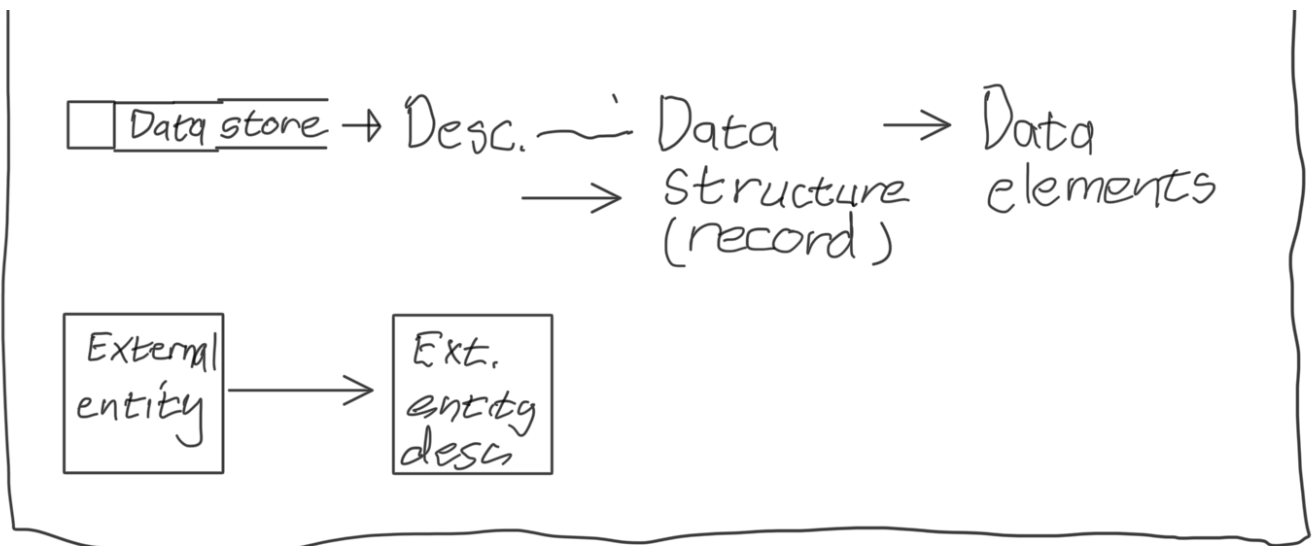
a noun

- Each process has name + reference number
- Context diag will have processes representing entire info sys
- If process has to be duplicated add *
- Each process must have name and unique reference number

1.3.4.2 Data dictionary

- Also called data repo.
 - Stores data about sys and org in central location
 - data flows
 - data stores
 - processes
 - entities
 - Defines data elements
 - field
 - data item
 - Smallest meaningful piece of data
 - describe data elements
 - Can group elements into meaningful combos
 - Think dashboard data
 - Record is combo of related elements
 - Included in dataflows
 - Stored in data stores
- eg title, author, price





- Items defined during structured analysis and creation of data dictionary shown above
- Data dictionary must document relationships above accurately to be consistent with DFD

Every element on the DFD must be documented in dictionary

- Can make notes on diagram and use CASE tools later, others enter data directly on usecase tool

Documenting data elements to be included in data dictionary

Name/label of element	Input/Output format
Aliases	Default value
Description of element	Prompt, field caption
Type	Source
Length	Security
Users (who input & change data)	- Id who has access to field.
Acceptable data	
Derivation formula	

Attributes of data flows that are documented

Attributes of data flows that are documented

DF Name	Derivation formula (If calculated)
Alias	
Origin (process, entity or data store)	
Dest	
Record (defined seperately, > 1 data flow & store can use same record)	

Attributes of data stores that are documented

name or label	output data flows
description	Record (name of record for data store)
Alt name	
input data flows	Volume & frequency

Documenting process

- Every data store must be documented
 - desc of atts of processes
 - Is it functional primitive
 - Process description
 - Model that describes business logic

Attributes of processes that need to be documented

- Name
- desc
- proc no
- input data flow
- output data flow
- Proc desc
 - Only filled if functional primitive

Documenting external entities

- ✓
 - Attributes
 - Data structure name
 - Alternate name
 - desc
 - input & output data flow

Documenting records

- Name
- Alt name
- desc
- Content or desc
 - List of elements making up record.
 - Primary key must be chosen