COMP 208

2020-21

Lecture 5

Documenting Design

Content

Details about:

Review Meeting

Design Process

Design Review Meeting

- This meeting reviews **the output** from the design stage
- The Design Review will result in the award of a score that will contribute to a maximum of **15%** of your team mark.

Organisation Details

- Design reviews will take place in week 6 ***(15 - 19 March)***.
- Full design reports to be submitted by ***Friday March 19th, by 12 noon through CANVAS.***
- ***Teams are responsible*** for arranging a time for the review
  - Send e-mail to ***reviewer***, cc-ed to all group members
  - Make your booking by ***Friday 12 March***.
- Reviews will typically last around 30 minutes.

Organisation Details

- Design reviews will take place in week 8 ***(15 - 19 March)***.
- Full design reports to be submitted by ***Friday March 19th by 12 noon through CANVAS.***
- ***Teams ar*** time for the rev
  - Send e- memb
  - Make y
- Reviews wil typ

50% of the mark for the report
50% for the performance in the review meeting

Failure to sign the report will result in the Student losing (at most) 50% of the mark
Failure to attend the meeting will result in the Student losing (at most) 50% of the mark

Form of the Review

- Agree with the Reviewer the **on-line platform** for the meeting (Teams, Zoom, BigBlueButton, Google meet, are some of the available options)
- Each team is expected to illustrate the main features of their design (full design to be included in the report).
- ***The reviewer*** may ask questions for clarification
- ***The reviewer*** may make constructive comments that should help the team improving their design

Feedback

- The online review meeting will normally be recorded
- A copy of the video will be returned to the team soon after the meeting
- As usual, a written feedback form will be returned in due course

DESIGN

Design??

Reality

Design??

Reality

HOW?

I know it! NO Design!

NO Design

Task: Write a Java program that prints the sentence

Hello `YOUR_NAME`! Today it's `date`.
Good morning!

(with the obvious substitution for the placeholders **`YOUR_NAME`** and **`date`**)

Design??

Reality

HOW?

Design??

Reality

Model

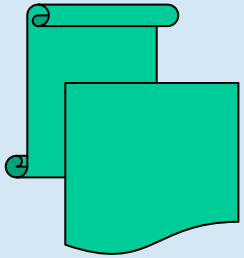
HOW?

Design??

HOW?

Design??

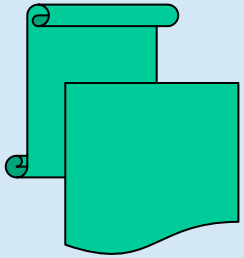
HOW?



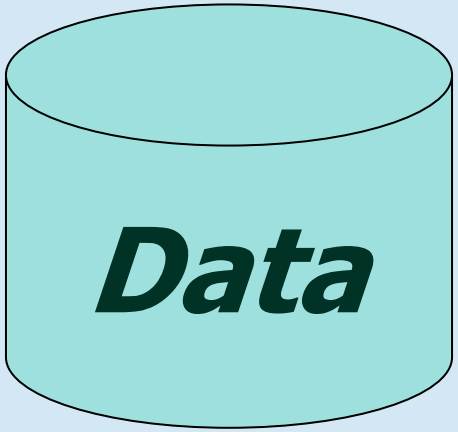
Processes

Design??

HOW?

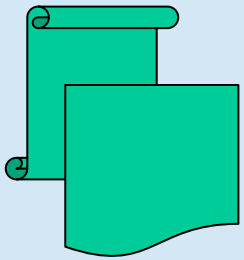


Processes

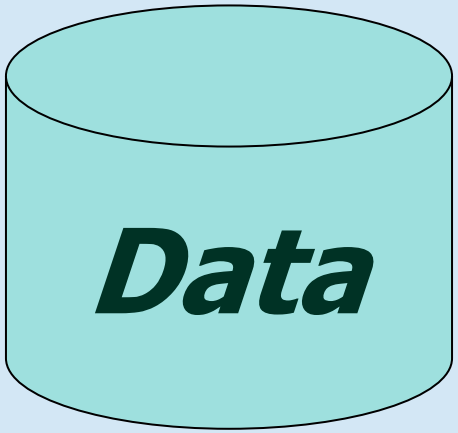


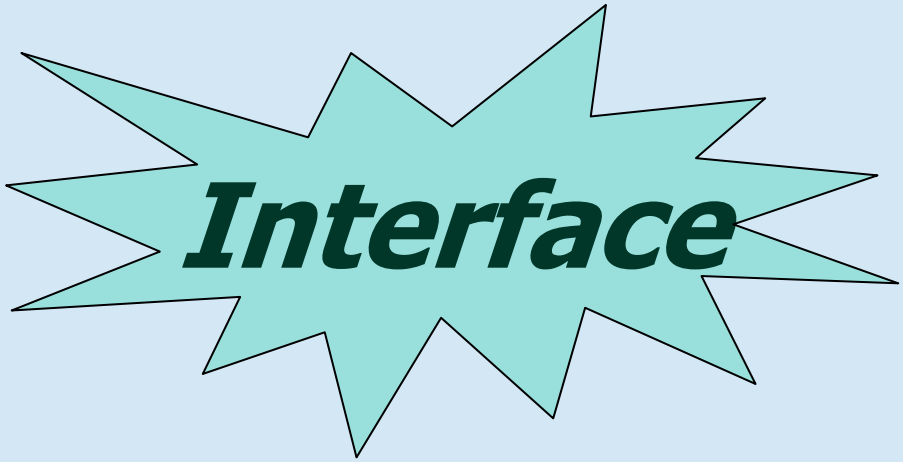
Design??

HOW?

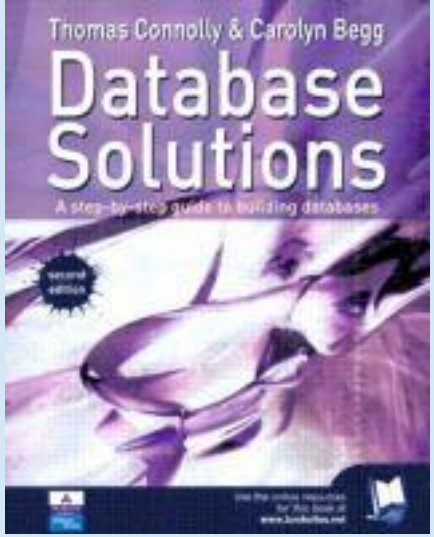


Processes

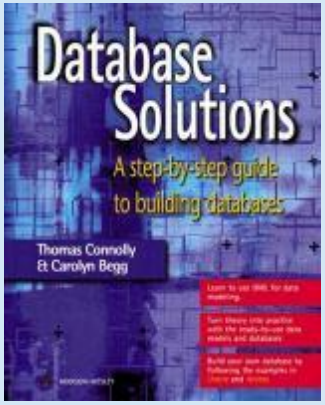


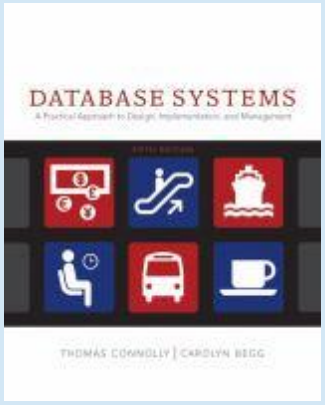


Database Design



- Connolly and Begg give a very detailed step by step guide to design
  - See Chapters 9-10 & 12-16 (Chapters 6-17 in 1st edition), summarised in Appendix B.
- Following this method produces a number of documents
- We review the most important of these.





Data(base) Design Method

- **Logical DB design**
  - Create and check the ***ER model***
  - (usually) Map the ER model to tables
- **Physical DB design**
  - Translate Logical *DB design for target DBMS*
  - Choose *file organization* and *indexes*
  - Design *user views*
  - Design *security mechanisms*
  - Design *controlled redundancy*
  - *Monitor* and *tune* operational system

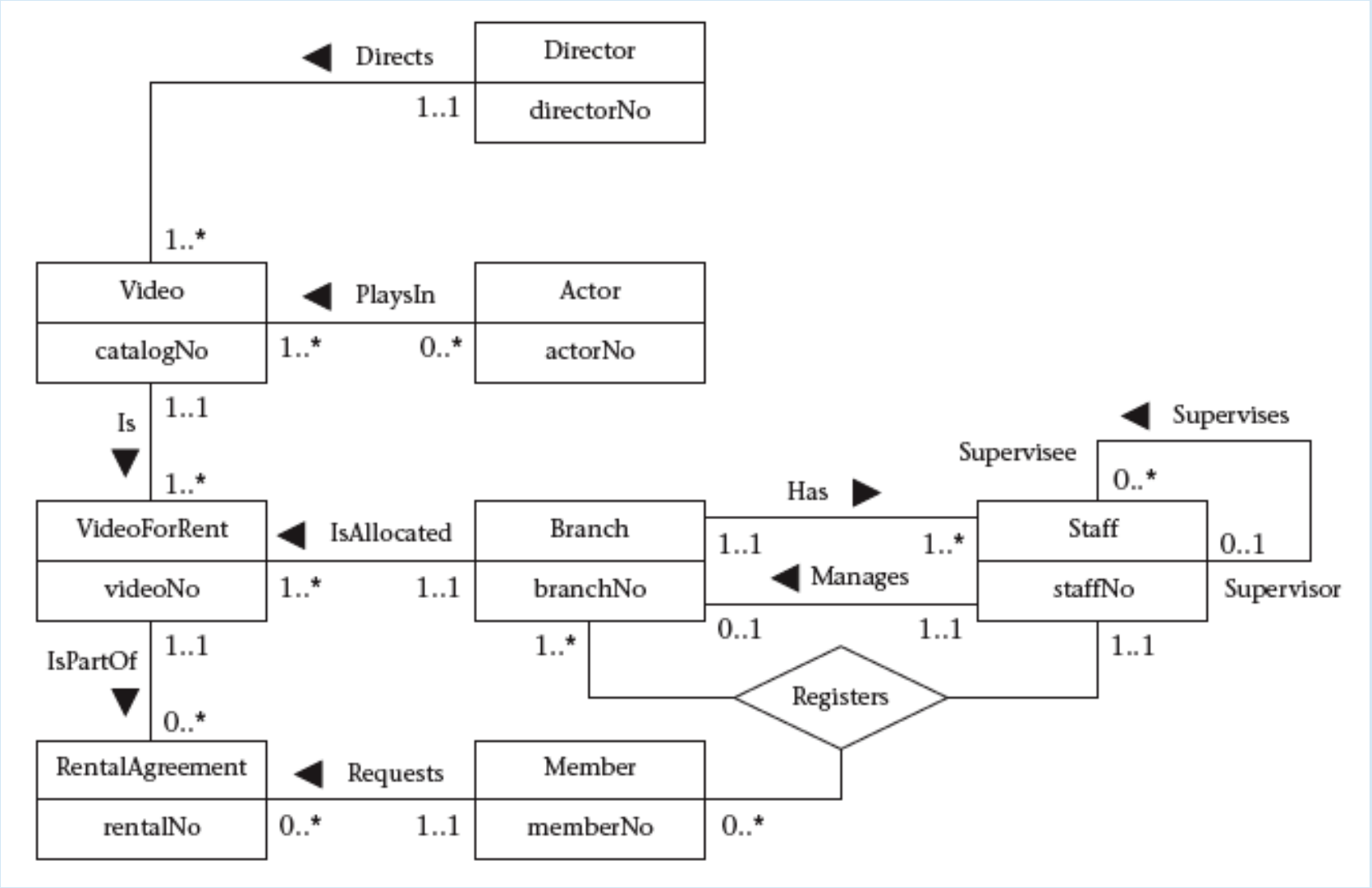
Data Model

- Logical model
- Dictionary
- Logical data structure organization
- Physical data structure organization
- Constraints
- Transactions

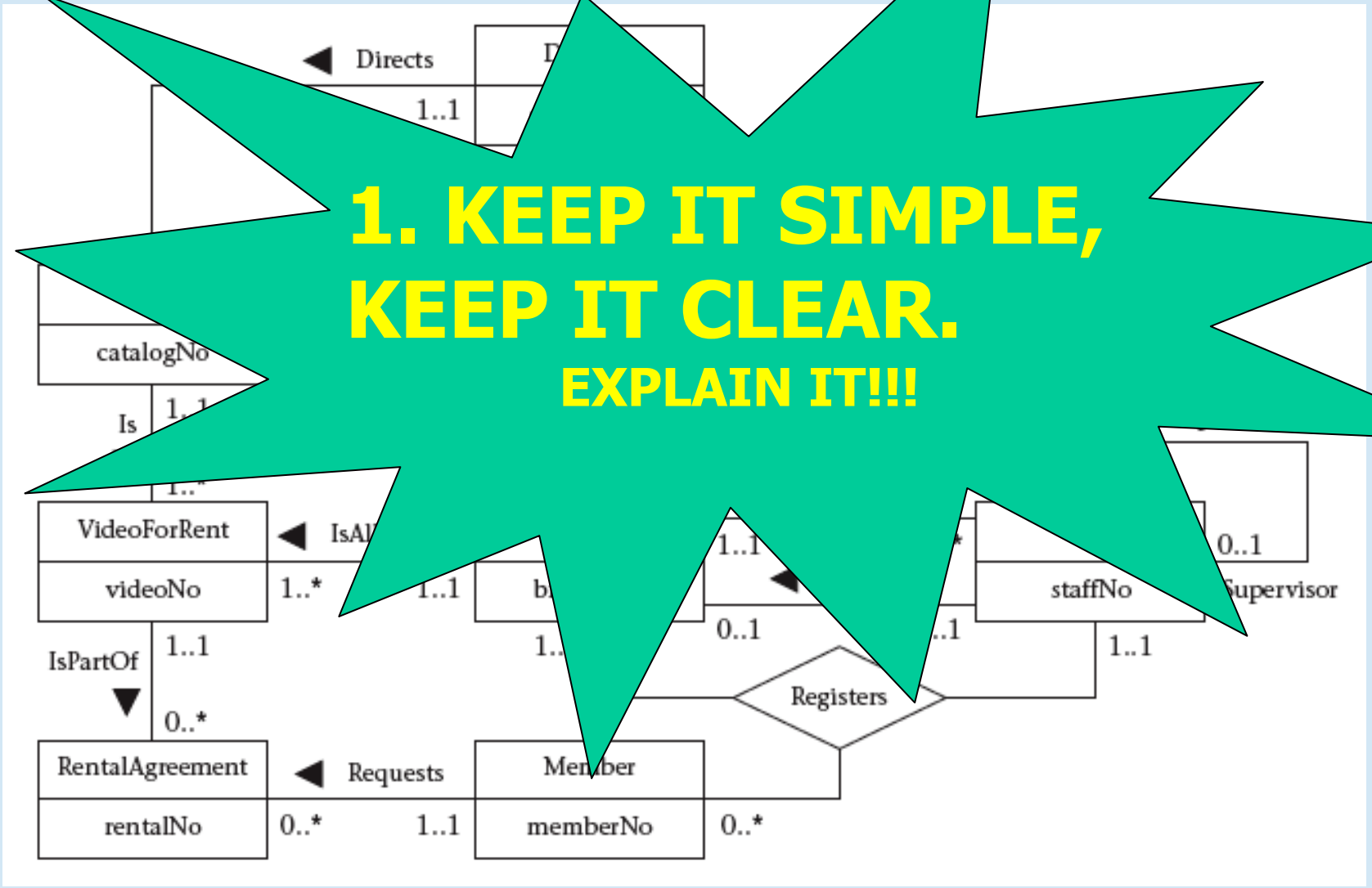
(1) (Global) Logical Data Model

- During design you may produce **local** data models for different user views
- These are brought together into a global data model. The ***global*** data model will be reviewed
- The global data model should be described in a diagram showing:
  - ***entities*** and their ***primary keys***,
  - ***relationships*** between entities and their ***direction*** and ***multiplicities***
- An example is given in:
  - Figure C.4 p 439 *(Figure 10.4 in 1st edition).*

Global ER diagram



Global ER diagram

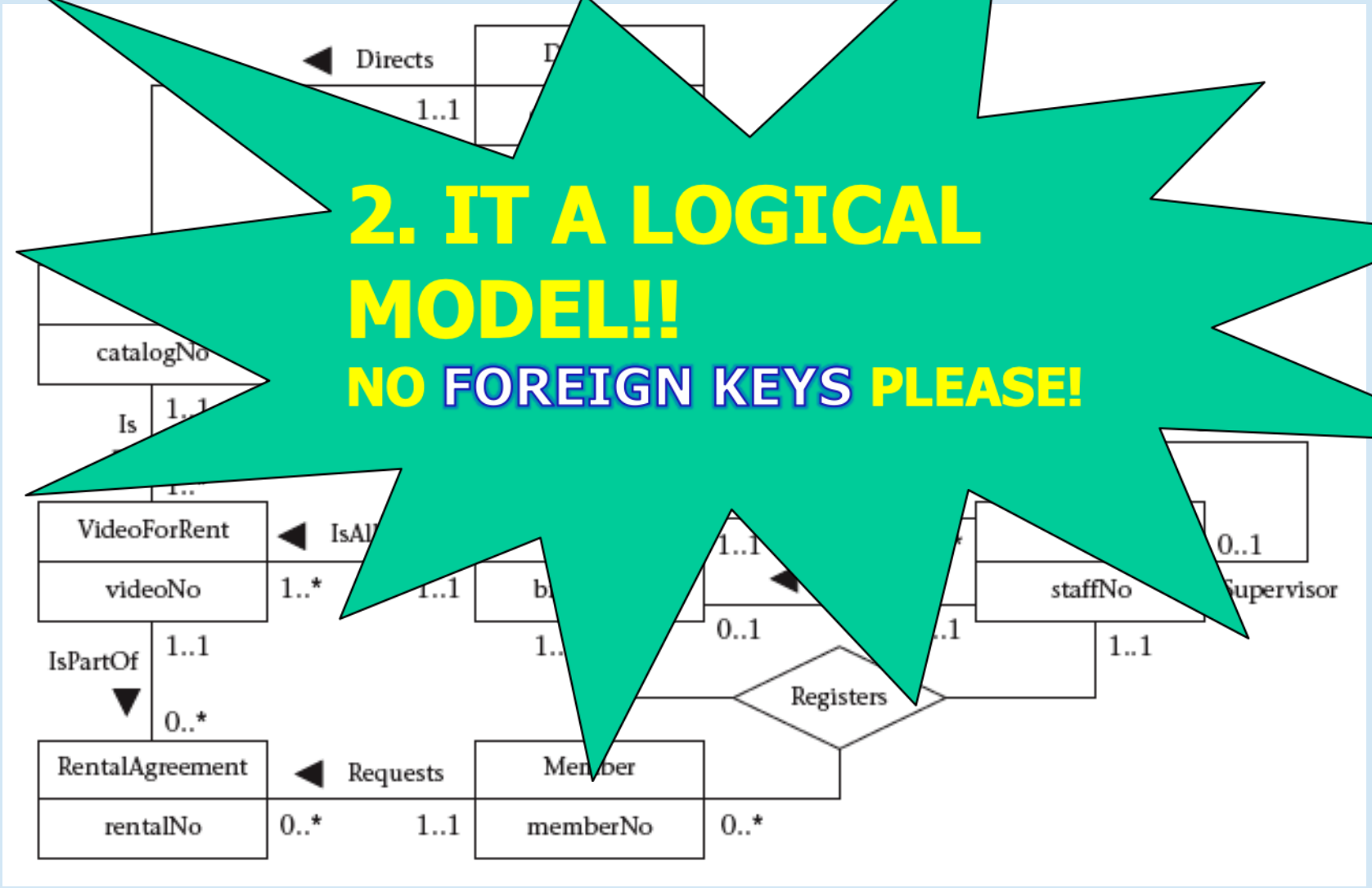


1. KEEP IT SIMPLE, KEEP IT CLEAR.

EXPLAIN IT!!!

KEYS?

Global ER diagram



(2) Data Dictionary

- The Data Dictionary is built up throughout the logical design phase
- It should contain all information about the data to be used by the system
- ***Entities*** require a description, any aliases used, and occurrence
  - See Fig 9.2 *(Fig 8.1 in 1st edition)*
- For each ***relationship*** in which an entity participates we need the multiplicity, the related entities and their multiplicity
  - See Fig 9.7 *(Fig 8.6 in 1st edition)*

Extract from data dictionary

Entity name	Description	Aliases	Occurrence
Branch	Place of work	Outlet and Branch Outlet	One or more *StayHome* branches are located in main cities throughout the US.
Staff	General term describing all staff employed by *StayHome*	Employee	Each member of staff works at a particular branch.

Extract from the data dictionary showing descriptions of relationships

Entity	Multiplicity	Relationship	Multiplicity	Entity
Branch	1..*	Has	1..1	Staff
Branch	1..*	IsAllocated	1..1	VideoForRent
Staff	0..1	Manages	1..1	Branch
Staff	0..*	Supervises	0..1	Staff

(2) Data Dictionary

- For each ***attribute*** of an ***entity***, a description, data type and length, whether nulls are allowed and whether it can be multi valued
  - See Fig 9.8 *(Fig 8.7 in 1st edition)*
- For each entity its primary and alternate ***keys***
  - Fig 9.10 *(Fig 8.9 in 1st edition)*
- Any ***integrity constraints*** on attributes resulting from referential integrity and business rule considerations
- Indication of ***derived data items***, and how they are computed
  - See p. 205 *(p. 139 in 1st edition).*

Sample of data dictionary showing descriptions of attributes

Entity	Attributes	Description	Data type and length	Nulls	Multi-valued	...
Branch	branchNo	Uniquely identifies a branch	4 fixed characters	No	No	
	address:					
	street	Street of branch address	30 variable characters	No	No	
	city	City of branch address	20 variable characters	No	No	
	state	State of branch address	2 fixed characters	No	No	
	zipCode	Zip code of branch address	5 variable characters	No	No	
	telNo	Telephone numbers of branch	10 variable characters	No	Yes	
Staff	staffNo	Uniquely identifies a member of staff	5 fixed characters	No	No	
	name	Name of staff member	30 variable characters	No	No	

Sample of data dictionary showing attributes with primary and alternate keys

Entity	Attributes	Description	Key	Nulls	...
Branch	branchNo	Uniquely identifies a branch	Primary key	No	
	address:				
	street	Street of branch address		No	
	city	City of branch address		No	
	state	State of branch address		No	
	zipCode	Zip code of branch address	Alternate key	No	
	telNo	Telephone numbers of branch		No	
Staff	staffNo	Uniquely identifies a member of staff	Primary key	No	
	name	Name of staff member		No	

(3) Logical Table Structure

- The Logical Table structures comprise, for each proposed table:
  - The ***name*** of the table
  - The ***columns*** for that table
  - the ***primary key*** for that table
  - any ***alternate keys*** for that table
  - any ***foreign keys*** for that table, and the tables they reference
- An example is given
  - Figure C.3 p. 438 *(Figure 10.5 in 1st edition).*

Tables for the global logical data model

Actor (actorNo, actorName) **Primary Key** actorNo	**Branch** (branchNo, street, city, state, zipCode, mgrStaffNo) **Primary Key** branchNo **Alternate Key** zipCode **Foreign Key** mgrStaffNo **references** Staff(staffNo)
Director (directorNo, directorName) **Primary Key** directorNo	**Member** (memberNo, fName, lName, address) **Primary Key** memberNo
Registration (branchNo, memberNo, staffNo, dateJoined) **Primary Key** branchNo, memberNo **Foreign Key** branchNo **references** Branch(branchNo) **Foreign Key** memberNo **references** Member(memberNo) **Foreign Key** staffNo **references** Staff(staffNo)	**RentalAgreement** (rentalNo, dateOut, dateReturn, memberNo, videoNo) **Primary Key** rentalNo **Alternate Key** memberNo, videoNo, dateOut **Foreign Key** memberNo **references** Member(memberNo) **Foreign Key** videoNo **references** Video(videoNo)
Role (catalogNo, actorNo, character) **Primary Key** catalogNo, actorNo **Foreign Key** catalogNo **references** Video(catalogNo) **Foreign Key** actorNo **references** Actor(actorNo)	**Staff** (staffNo, name, position, salary, branchNo, supervisorStaffNo) **Primary Key** staffNo **Foreign Key** branchNo **references** Branch(branchNo) **Foreign Key** supervisorStaffNo **references** Staff(staffNo)
Supplier (supplierNo, name, address, telNo, status) **Primary Key** supplierNo **Alternate Key** telNo	**Telephone** (telNo, branchNo) **Primary Key** telNo **Foreign Key** branchNo **references** Branch(branchNo)
Video (catalogNo, title, category, dailyRental, price, directorNo, supplierNo) **Primary Key** catalogNo **Foreign Key** directorNo **references** Director(directorNo) **Foreign Key** supplierNo **references** Supplier(supplierNo)	**VideoForRent**(videoNo, available, catalogNo, branchNo) **Primary Key** videoNo **Foreign Key** catalogNo **references** Video(catalogNo) **Foreign Key** branchNo **references** Branch(branchNo)
VideoOrder (orderNo, dateOrdered, dateReceived, branchNo) **Primary Key** orderNo **Foreign Key** branchNo **references** Branch(branchNo)	**VideoOrderLine** (orderNo, catalogNo, quantity) **Primary Key** orderNo, catalogNo **Foreign Key** orderNo **references** VideoOrder(orderNo) **Foreign Key** catalogNo **references** Video(catalogNo)

(4) Physical Table Structure

- The physical table structures comprise, for each table to be implemented,
  - the name of the table,
  - the domains of the columns
  - the column names, their domains, and whether they may be null
  - the primary and any alternate keys
  - any foreign keys and their associated integrity constraints
- An example for one physical table is given at Figure 12.2 *(both editions).*

Example of Physical Table

```
domain Branch_Numbers       fixed length character string length 4
domain Street_Names         variable length character string maximum length 30
domain City_Names           variable length character string maximum length 20
domain State_Codes          fixed length character string length 2
domain Zip_Codes            fixed length character string length 5
domain Staff_Numbers        fixed length character string length 5

Branch(  branchNo      Branch_Numbers      NOT NULL,
         street        Street_Names        NOT NULL,
         city          City_Names          NOT NULL,
         state         State_Names         NOT NULL,
         zipCode       Zip_Codes           NOT NULL,
         mgrStaffNo    Staff_Numbers       NOT NULL)
         Primary Key branchNo
         Alternate Key zipCode
         Foreign Key mgrStaffNo References Staff(staffNo) ON UPDATE CASCADE ON DELETE NO ACTION
```

(5) Business Rules

- Business Rules express constraints on data that can be entered
  - e.g. 10 books for students, 30 books for staff may be borrowed from a library
- They are instantiated **policies** of the client company or organization expressed in the form of rules over the data
  - Note that Business Rules are more precise than company policies
- Documentation should show any such rules and how they will be implemented
  - validation rules on fields
  - validation rules for records.

Example of Policies and Rules

- Business Policies
  - Only registered students may use the University Library.
  - Students may only borrow a limited number of books at any one time.
- Business Rules
  - Each student may only borrow 10 books at any one time.
- Business Rules force constraints on the data in the database
  - eg, the number of books linked to each student name in the library database is not more than 10.
- Not all Business Policies are translated into Business Rules.

(6) Transaction Table/Matrix

- The transaction table/matrix shows, for each transaction, what tables are used, and how they are used
- Each table forms a row
- Each transaction has four columns
  - insert
  - read
  - update
  - delete
- An example is given as Table 13.1 *(both editions).*

Transactions and tables matrix

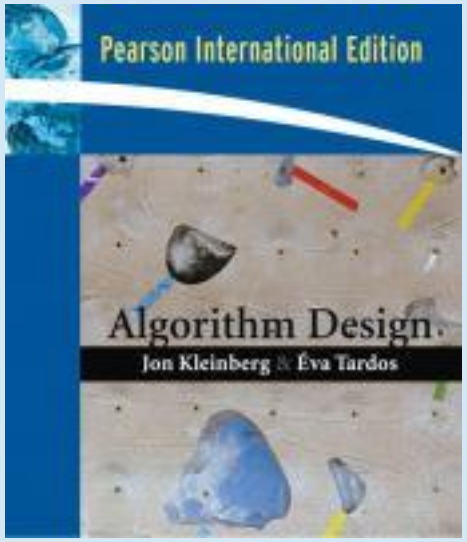
Transaction (e): Enter details of new member registering at a branch
Transaction (k): Update/delete the details of a given member
Transaction (p): List the title, category and availability of all videos at a specific branch.
etc.

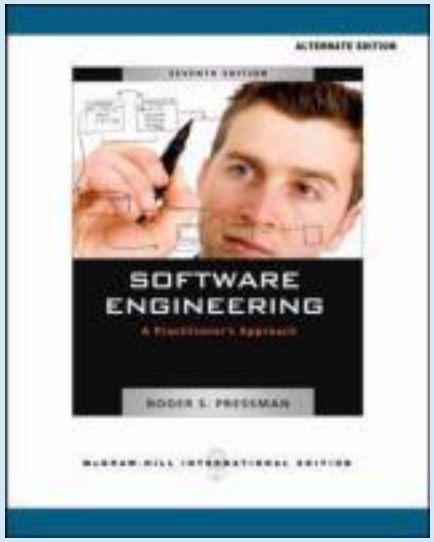
Table 13.1 Cross-referencing transactions and tables.

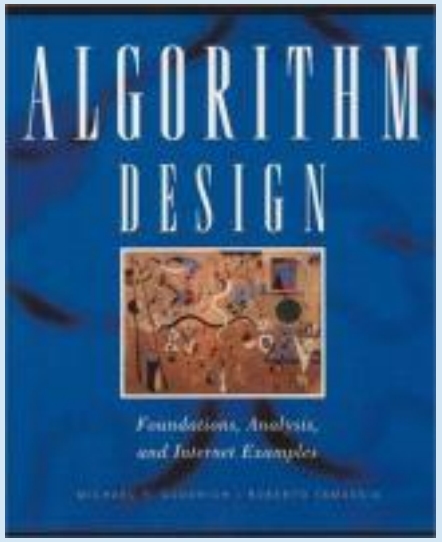
Transaction/ Table	(e)				(k)				(p)				(q)				(r)				(s)			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
Branch																								
Staff		X																						
Video										X				X				X				X		
VideoForRent										X				X				X				X		
RentalAgreement																						X		
Member	X					X	X	X														X		
Registration	X																							
Actor														X										
Role														X										
Director																		X						

I = Insert; R = Read; U = Update; D = Delete

Process Design





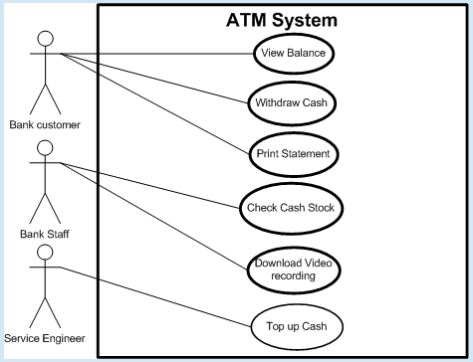


Process Design

Some possible design tools:

- Use-cases
  - Descriptions of typical usage situations
- Data flow diagrams
  - Showing which data items are transferred between which components, in which scenarios
- Navigation path diagrams
- Storyboards
- Functional descriptions of components
- Pseudo-code, flow charts, sequence/activity diagrams

Use Cases and Sequence Diagrams



- Use Cases show what happens when each type of user interacts with the system
  - By recording all the ways the system is used ("cases of use") we accumulate all the goals or requirements of the system
- The Use Case is a collection of sequences of actions or events relating to a particular goal
- We would typically develop use cases for all the main processes in *normal* operation, and for many of the processes in *abnormal* operation.

Functional Descriptions

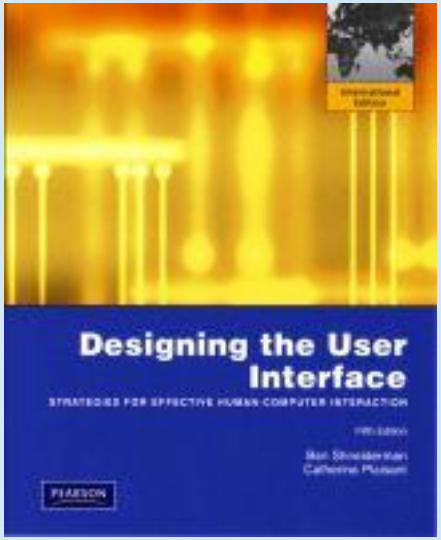
A Functional description provides details about the function of a component, along with actions, pre-conditions and post-conditions associated with that component.

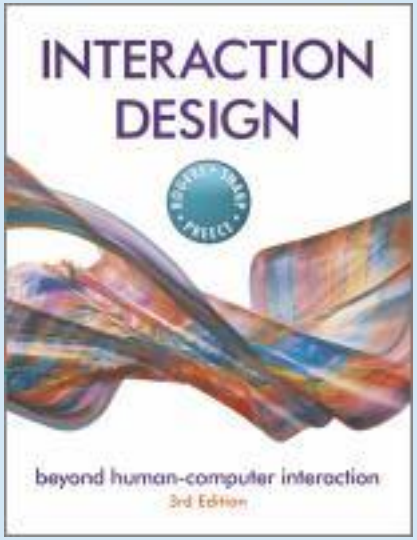
Example:

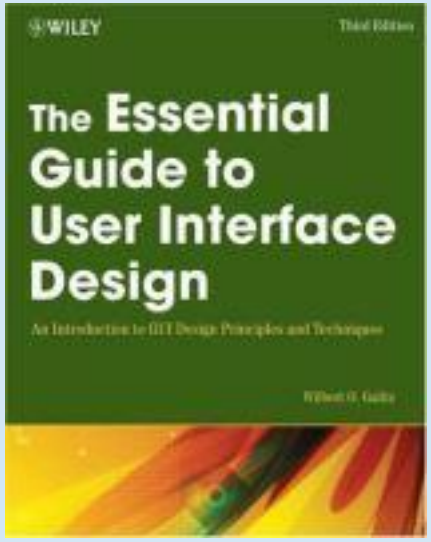
Stock manager function in Book Inventory system

- Description: The function manages the stock of books available as books enter and leave the book database, ordering new stock as needed. It maintains information as to when stock is expected to arrive.
- Goals: Re-order books, log books arriving, log books outgoing.
- Actions: Send e-mail to order new stock.
- Triggers: Stock arrival, stock order delay, failed stock arrival.
- Information used: Stock database, Customer order, Stock order.
- Information produced: Stock database, Delayed orders, Arrived orders.

Interface Design







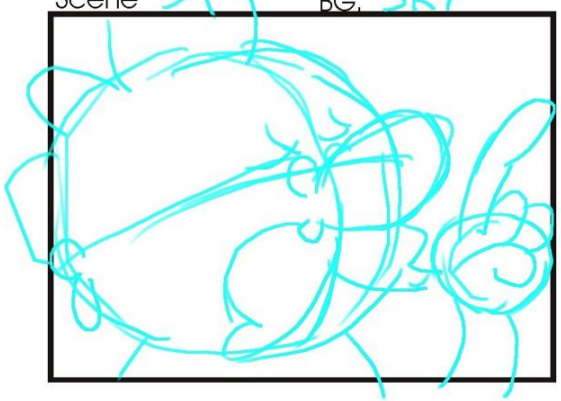
Title: Dr floyd 4b8

Scene 30 BG. cont

ACTION

DIAL. DR. FLOYD: UH...YEAH...
OR,

Scene 31 BG. sky

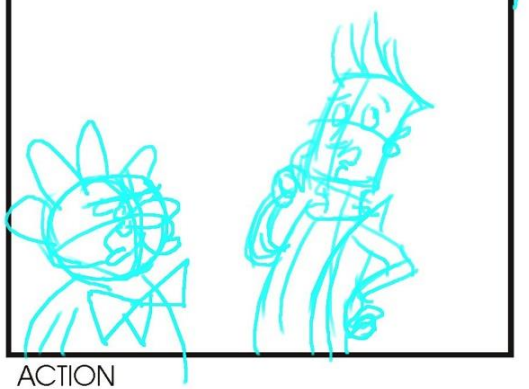


WE COULD POSE AS ARTISTS AND ASK TO PAINT HIS PORTRAIT.

Scene 32 BG. cont

HEY THAT'S A GREAT IDEA TOO!

Scene 33 BG. grass/sky



ACTION

DIAL. NOW, WHICH ONE DO YOU THINK WE SHOULD TRY?

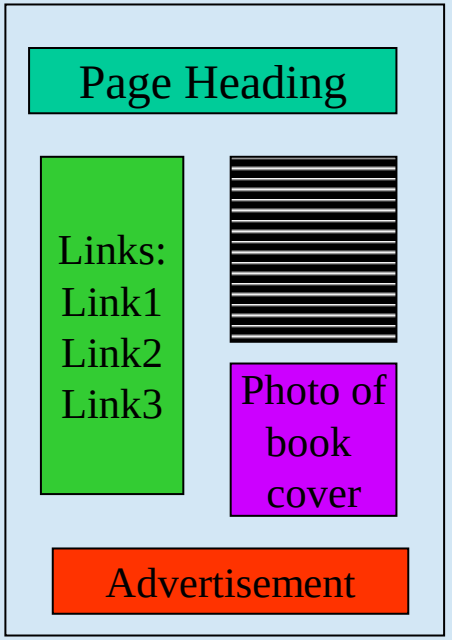
Scene 34 BG. sky

DR. FLOYD: WE SHOULD POSE AS ARTISTS AND ASK TO PAINT HIS PORTRAIT?
DR. GRANT: OH OKAY,

Scene 35 BG. sky/grass

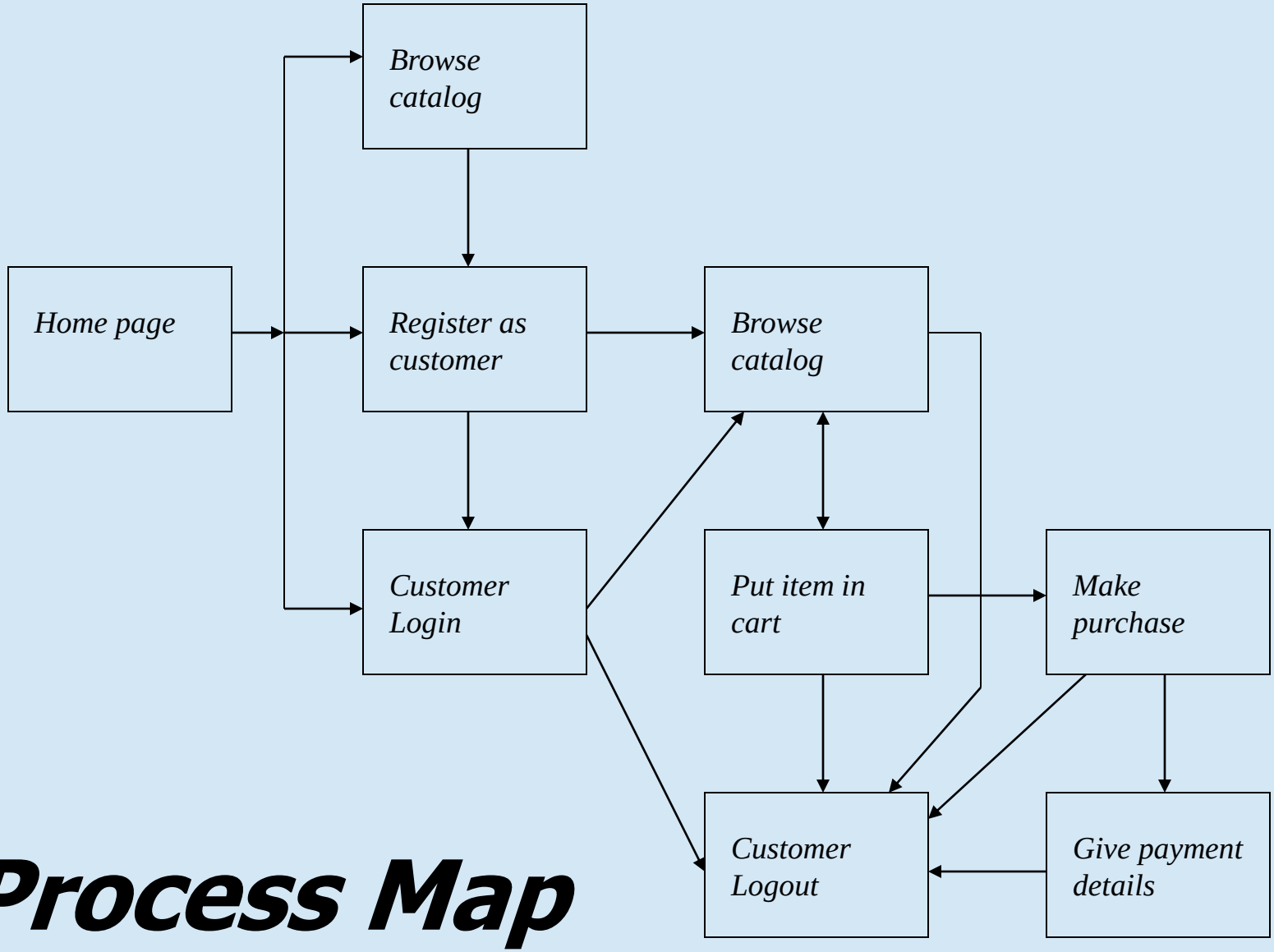
WE'LL DO IT
YOUR WAY THIS TIME.

Storyboards

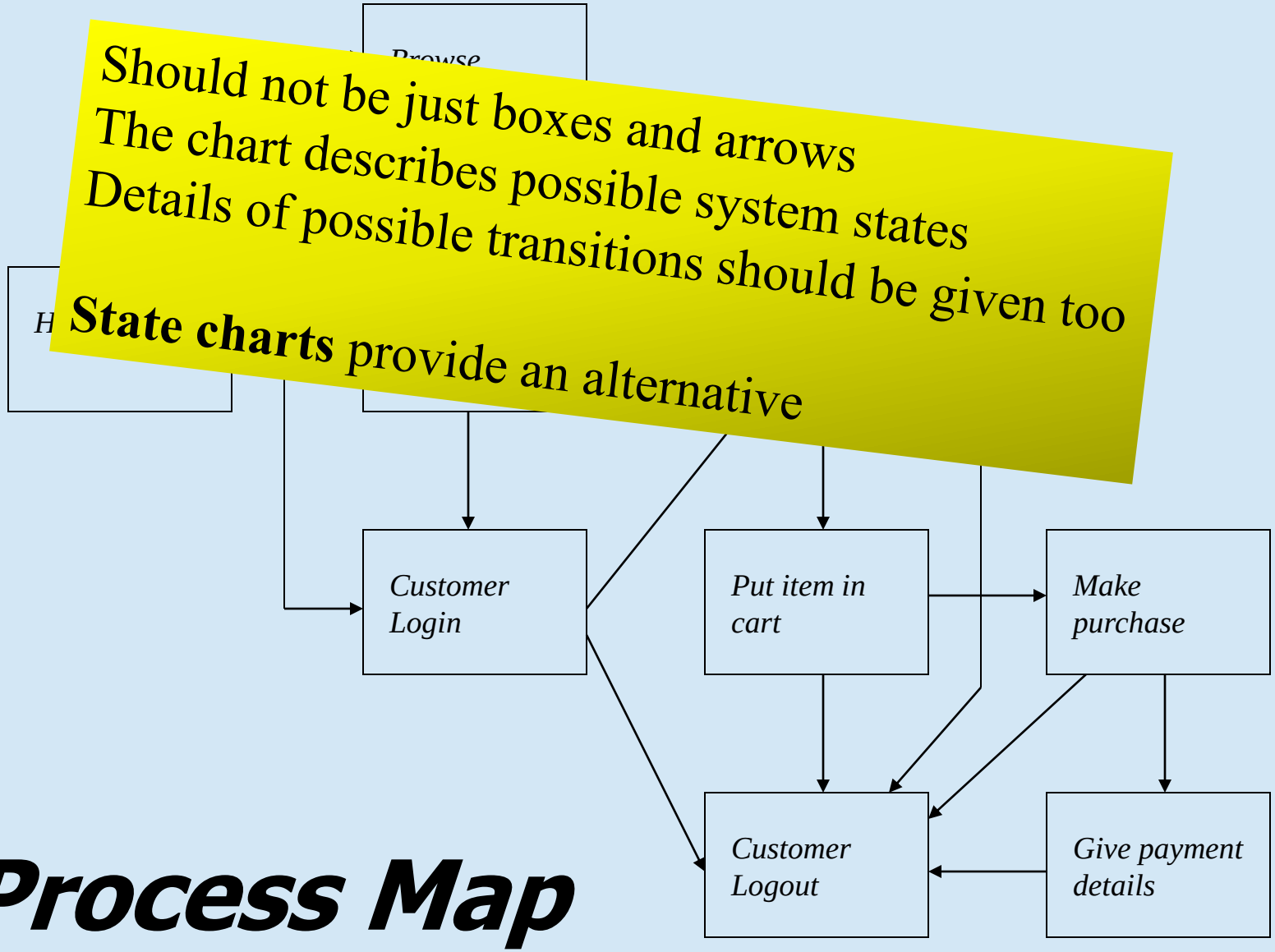


- These give an outline of each web-site, showing:
  - The structure of each web-page
    - Where the text is
    - Where the graphics are, and what these will consist of
    - Where the links are
  - The flow of control between pages
- The term comes from the movie industry (check Wiki page)
  - Before a movie is made, each scene is mapped onto story boards to enable
    - Planning of rehearsals and shooting
    - Obtaining props and costumes
    - Planning of lighting, sound, camera angles, etc.

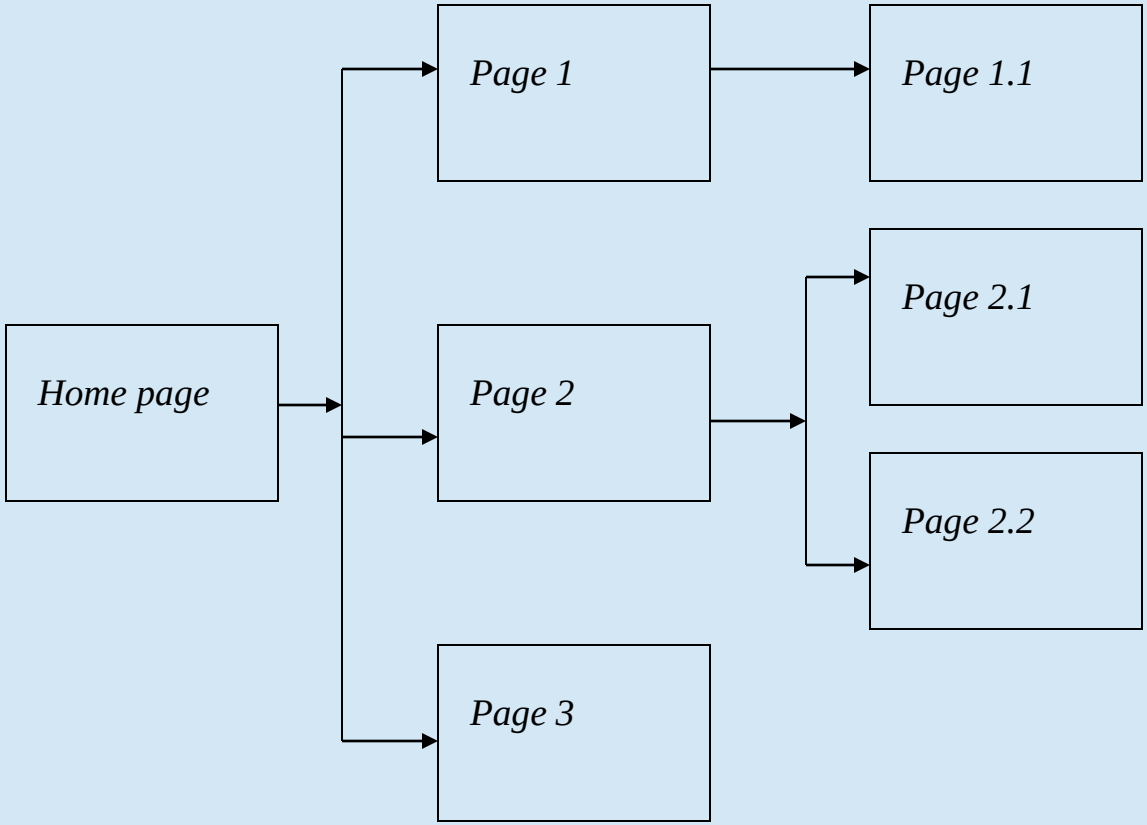
A Process Map



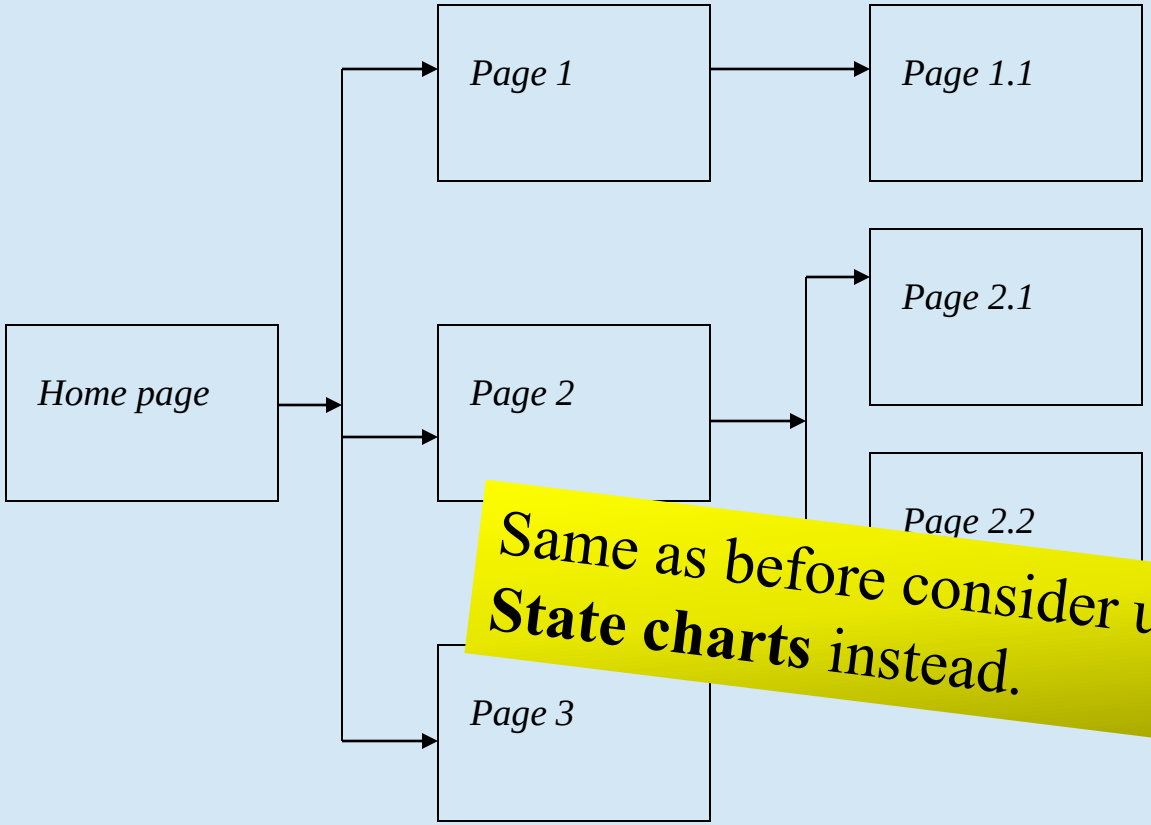
A Process Map



Navigation structure charts



Navigation structure charts



Summary

Design specifies

HOW?

Process design

Data design

Interface design

Output: Design documents

Summary (references)

Design specifies

HOW?

Process design (SE, Algorithms, Data Structure)

Data design (DB courses)

Interface design (Human Centric Computing)

Output: Design documents

Further Issues

- Class Diagrams
  - For non-DB applications, they replace the ER model
  - For DB applications, the could complement the conceptual data model
- Lots of diagrams/tools, writing well is hard.
- Description of general system dynamics
  - Often missing …

Design and Plan

- Don’t forget to include details about your planning in your documentation and in your presentation
  - This should show your progress so far and how it all worked, and your plans for the remainder of the project.
  - In particular, plans to the implementation stage should be quite clear and detailed.