

Process Modeling



Chapter 6

(with additions by Yale Braunstein)

Key Definitions



- ☑ A *process model* is a formal way of representing how a business operates
- ☑ *Data flow diagramming* shows business processes and the data that flows between them

Key Definitions

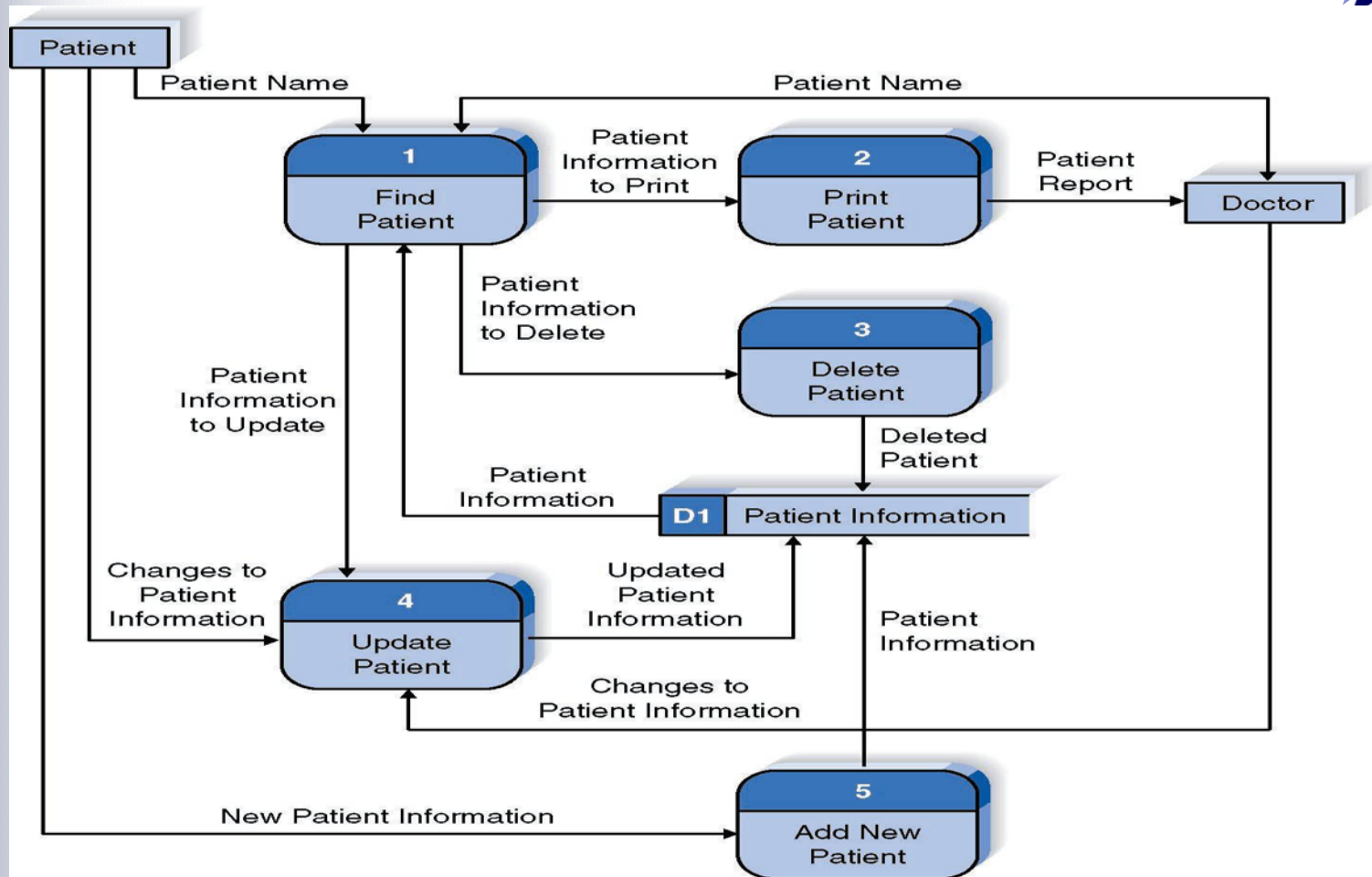


- ☑ *Logical process* models describe processes without suggesting how they are conducted
- ☑ *Physical models* include information about how the processes are implemented









Data Flow Diagrams



Reading a DFD



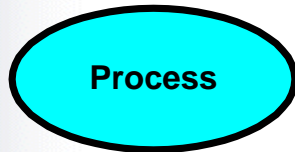
DFD Elements

Data Flow Diagram Element	Typical Computer-Aided Software Engineering Fields	Gane and Sarson Symbol	DeMarco and Yourdan Symbol
<p>Every <i>process</i> has</p> <ul style="list-style-type: none"> A number A name (verb phrase) A description One or more output data flows Usually one or more input data flows 	<p>Label (name)</p> <p>Type (process)</p> <p>Description (what is it)</p> <p>Process number</p> <p>Process description (Structured English)</p> <p>Notes</p>		
<p>Every <i>data flow</i> has</p> <ul style="list-style-type: none"> A name (a noun) A description One or more connections to a process 	<p>Label (name)</p> <p>Type (flow)</p> <p>Description</p> <p>Alias (another name)</p> <p>Composition (description of data elements)</p> <p>Notes</p>		
<p>Every <i>data store</i> has</p> <ul style="list-style-type: none"> A number A name (a noun) A description One or more input data flows Usually one or more output data flows 	<p>Label (name)</p> <p>Type (store)</p> <p>Description</p> <p>Alias (another name)</p> <p>Composition (description of data elements)</p> <p>Notes</p>		
<p>Every <i>external entity</i> has</p> <ul style="list-style-type: none"> A name (a noun) A description 	<p>Label (name)</p> <p>Type (entity)</p> <p>Description</p> <p>Alias (another name)</p> <p>Entity description</p> <p>Notes</p>		

DFD Shapes from Visio

Visio 5.x

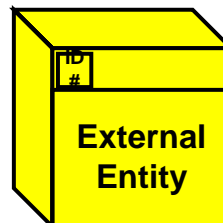
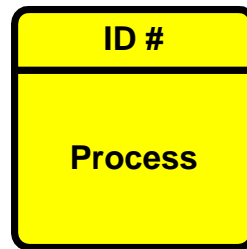
From Flow Chart /
Data Flow Diagram



Data Store

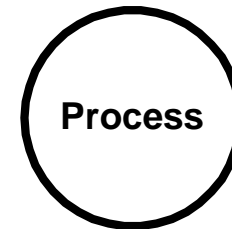


From Software Diagram /
Gane-Sarson DFD

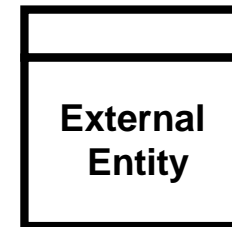


Visio 2000

From Flow Chart /
Data Flow Diagram

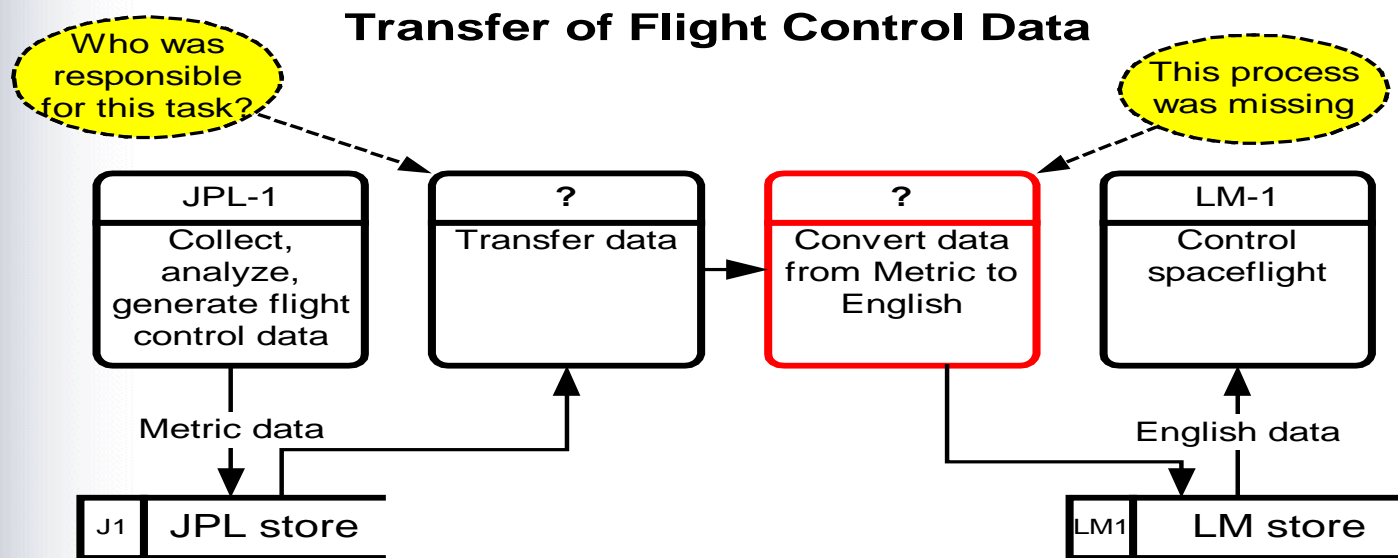


Data Store



DFD – Practical Example

Launched Dec. 11, 1998, the Climate Orbiter plunged too steeply into the Martian atmosphere Sept. 23, 1999, and either burned up or crashed. In an initial failure report released Oct. 15, 2000 the review board blamed the navigation error on a communications foul-up between NASA's Jet Propulsion Laboratory and prime contractor Lockheed Martin.



Structured English

Common Statements	Example
Action Statement	Profits = Revenues - Expenses Generate Inventory - Report Add Product record to Product Data Store
If Statement	IF Customer Not in Customer Data Store THEN Add Customer record to Customer Data Store ELSE Add Current-Sale to Customer's Total-Sales Update Customer record in Customer Data Store
For Statement	FOR all Customers in Customer Data Store Generate a new line in the Customer-Report Add Customer's Total-Sales to Report-Total
Case Statement	CASE If Income < 10,000: Marginal-tax-rate = 10% If Income < 20,000: Marginal-tax-rate = 20% If Income < 30,000: Marginal-tax-rate = 31% If Income < 40,000: Marginal-tax-rate = 35% ELSE Marginal-tax-rate = 38% ENDCASE

Key Definition



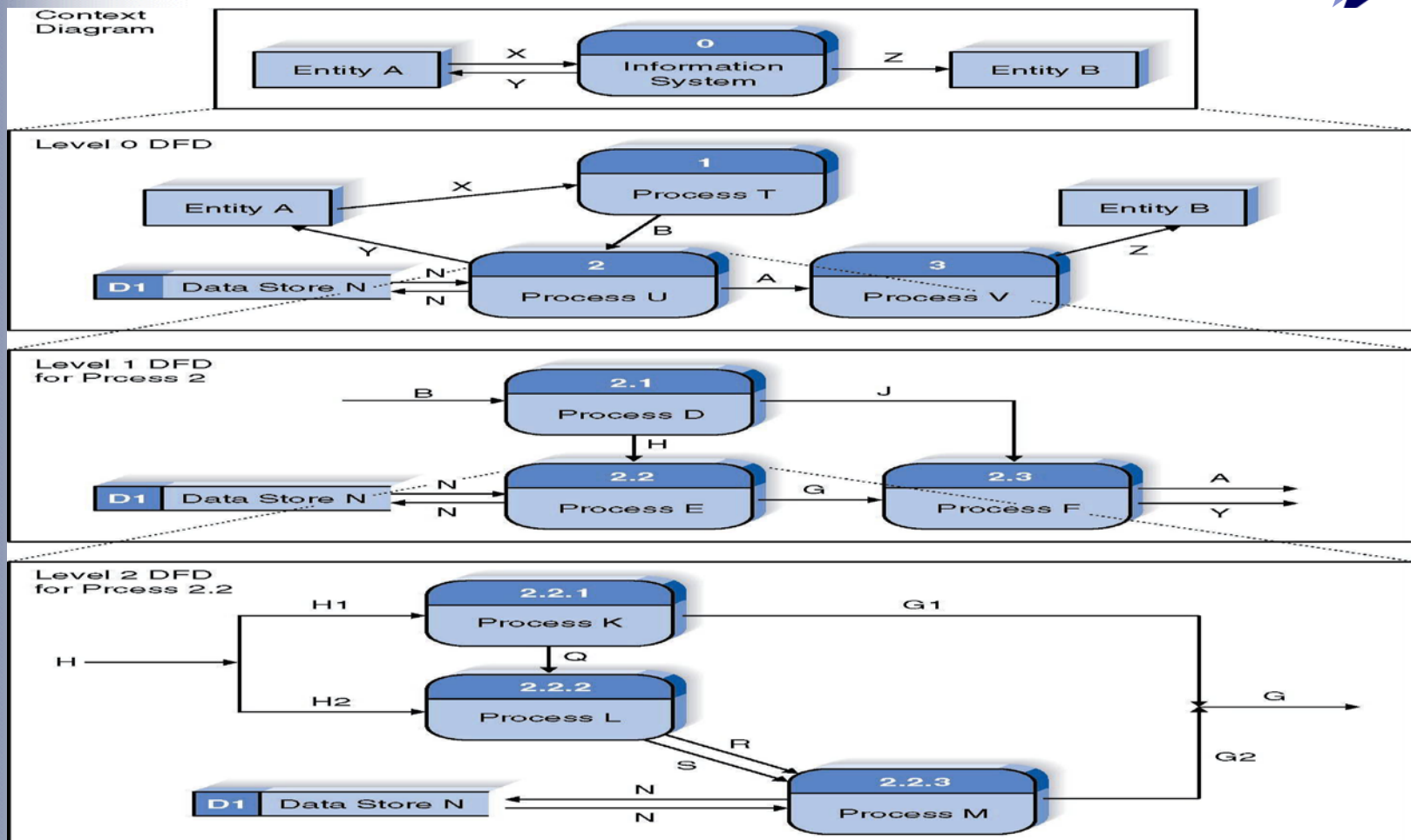
- ☑ *Decomposition* is the process of modeling the system and its components in increasing levels of detail.
- ☑ *Balancing* involves insuring that information presented at one level of a DFD is accurately represented in the next level DFD.

Context Diagram

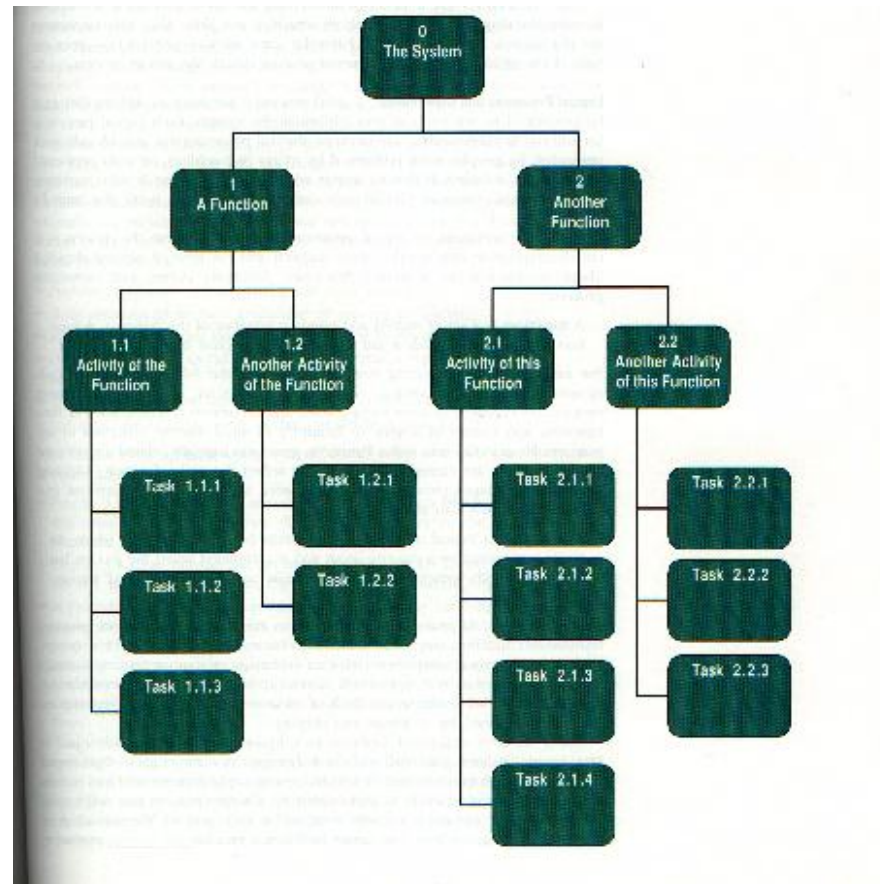


- ☑ Shows the context into which the business process fits
- ☑ Shows the overall business process as just *one* process
- ☑ Shows all the outside entities that receive information from or contribute information to the system

Relationship Among DFD levels



Decomposition Diagram



Level 0 Diagram



- ☑ Shows all the processes that comprise the overall system
- ☑ Shows how information moves from and to each process
- ☑ Adds data stores

Level 1 Diagrams



- ❑ Shows all the processes that comprise a single process on the level 0 diagram
- ❑ Shows how information moves from and to each of these processes
- ❑ Shows in more detail the content of higher level process
- ❑ Level 1 diagrams may not be needed for all level 0 processes

Level 2 Diagrams



- ☑ Shows all processes that comprise a single process on the level 1 diagram
- ☑ Shows how information moves from and to each of these processes
- ☑ Level 2 diagrams may not be needed for all level 1 processes
- ☑ Correctly numbering each process helps the user understand where the process fits into the overall system

Data Flow Splits and Joins



- ❑ A data flow split shows where a flow is broken into its component parts for use in separate processes
- ❑ Data flow splits need not be mutually exclusive nor use all the data from the parent flow
- ❑ As we move to lower levels we become more precise about the data flows
- ❑ A data flow join shows where components are merged to describe a more comprehensive flow

Alternative Data Flows



- ☑ Where a process can produce different data given different conditions
- ☑ We show both data flows and use the process description to explain why they are alternatives
- ☑ Tip -- alternative data flows often accompany processes with IF statements

Your Turn



- ☑ At this point in the process it is easy to lose track of the “big picture”.
- ☑ Describe the difference between data flows, data stores, and processes.
- ☑ Describe in your own words the relationship between the DFD and the ultimate new application being developed.

Creating Use Cases



Elements of a Use Case



- ☑ Trigger -- event that causes the scenario to begin
 - ▣ External trigger
 - ▣ Temporal trigger
- ☑ All possible inputs and outputs
- ☑ Individual steps
 - ▣ Show sequential order
 - ▣ Show conditional steps

Scenario Template (Use Case)

Scenario Description

Scenario Name: _____		ID Number: _____	
Short Description: _____			
Trigger: _____			
Type: External / Temporal			
Major Inputs:		Major Outputs:	
Description	Source	Description	Destination
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
Major Steps Performed		Information for Steps	
_____		_____	

This template can be downloaded from the [course download page](#).

Creating Data Flow Diagrams



Integrating Scenario Descriptions



- ☑ DFDs generally integrate scenario descriptions
- ☑ Names of use cases become processes
- ☑ Names of inputs and outputs become data flows
- ☑ Combining “small” data inputs and outputs into a single flow

Steps in Building DFDs



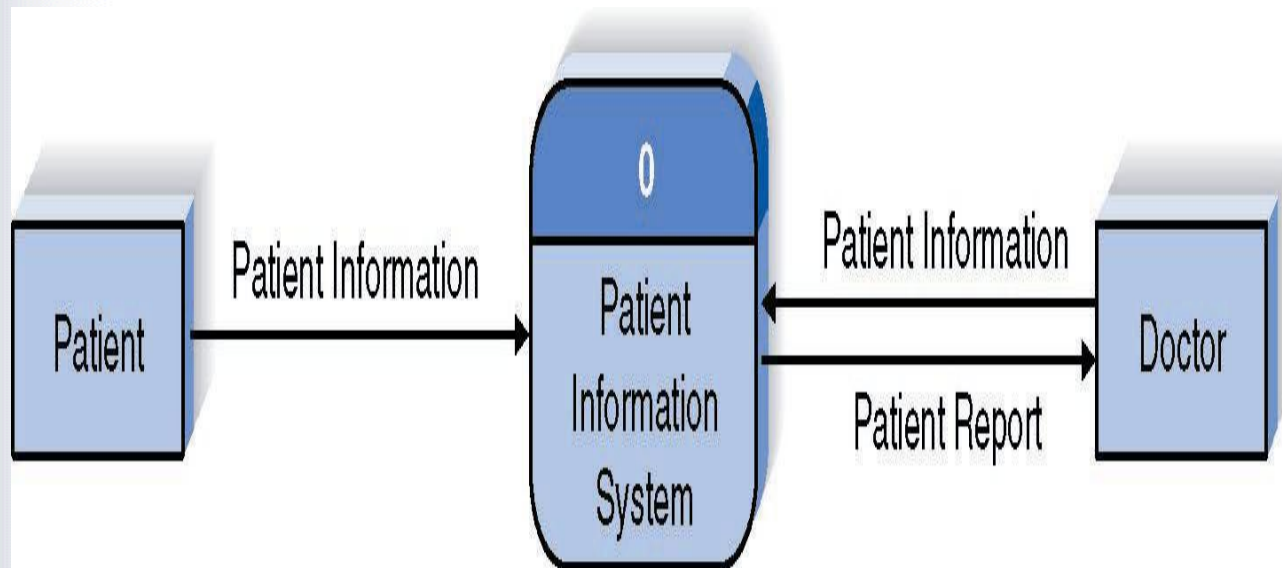
- ☑ Build the context diagram
- ☑ Create DFD fragments for each scenario
- ☑ Organize DFD fragments into level 0
- ☑ Decompose level 0 DFDs as needed
- ☑ Validate DFDs with user

DFD Fragment Tips

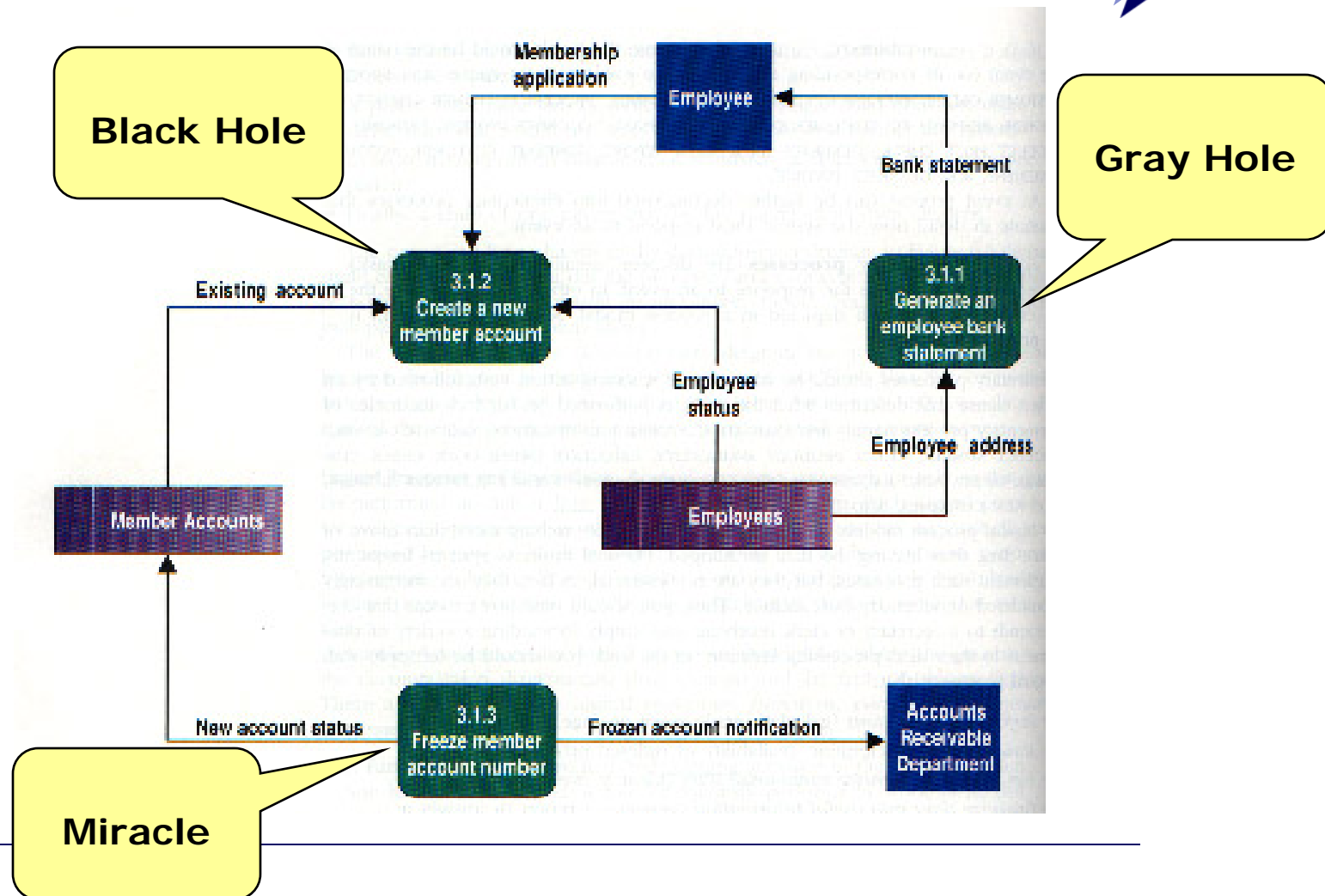


- ☑ All process names must be verb phrases
- ☑ Maintain organization's viewpoint in naming processes
- ☑ Layouts often place
 - ▢ processes in the center
 - ▢ inputs from the left
 - ▢ outputs to the right
 - ▢ stores beneath the processes

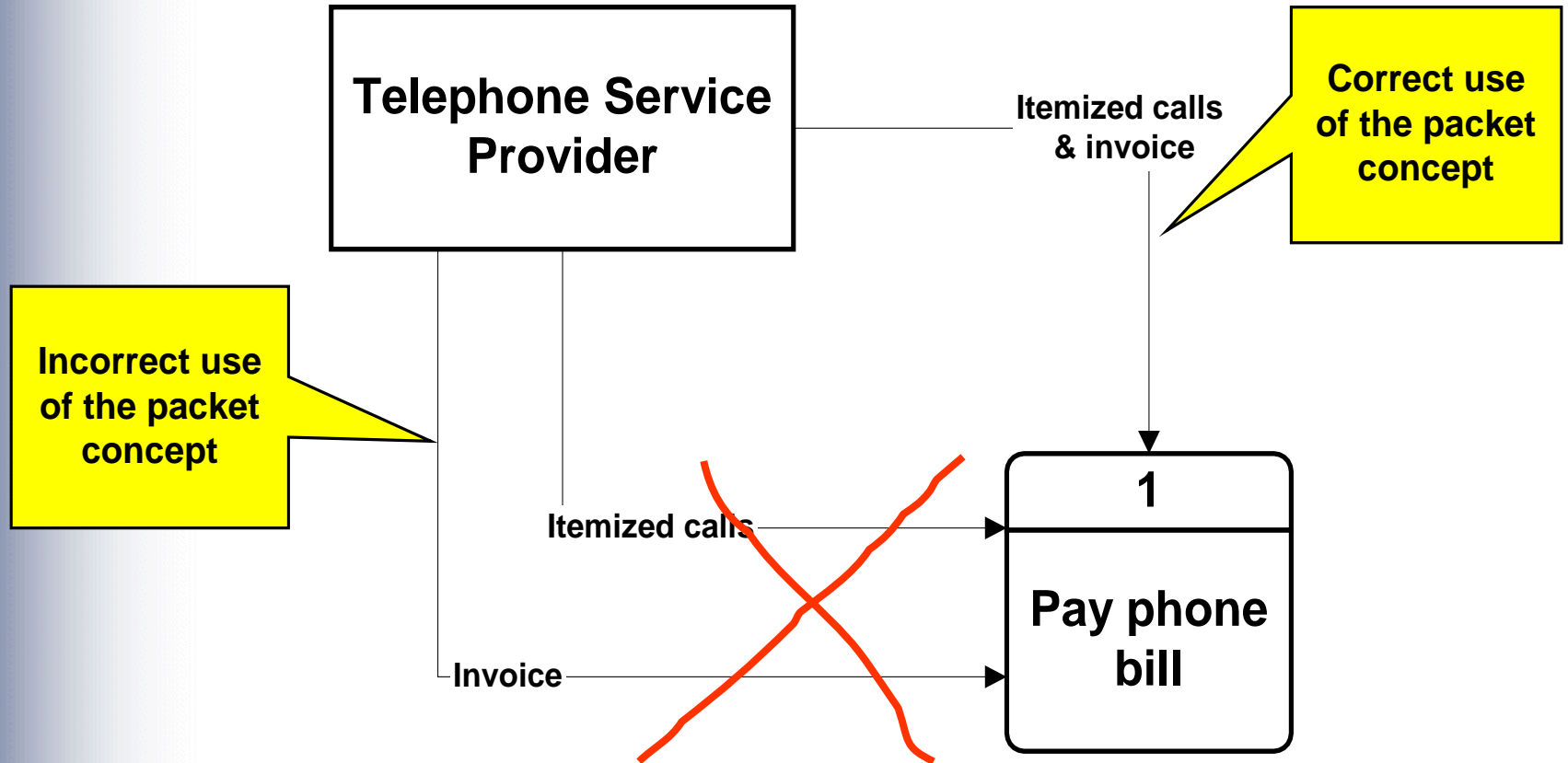
A DFD Fragment Example



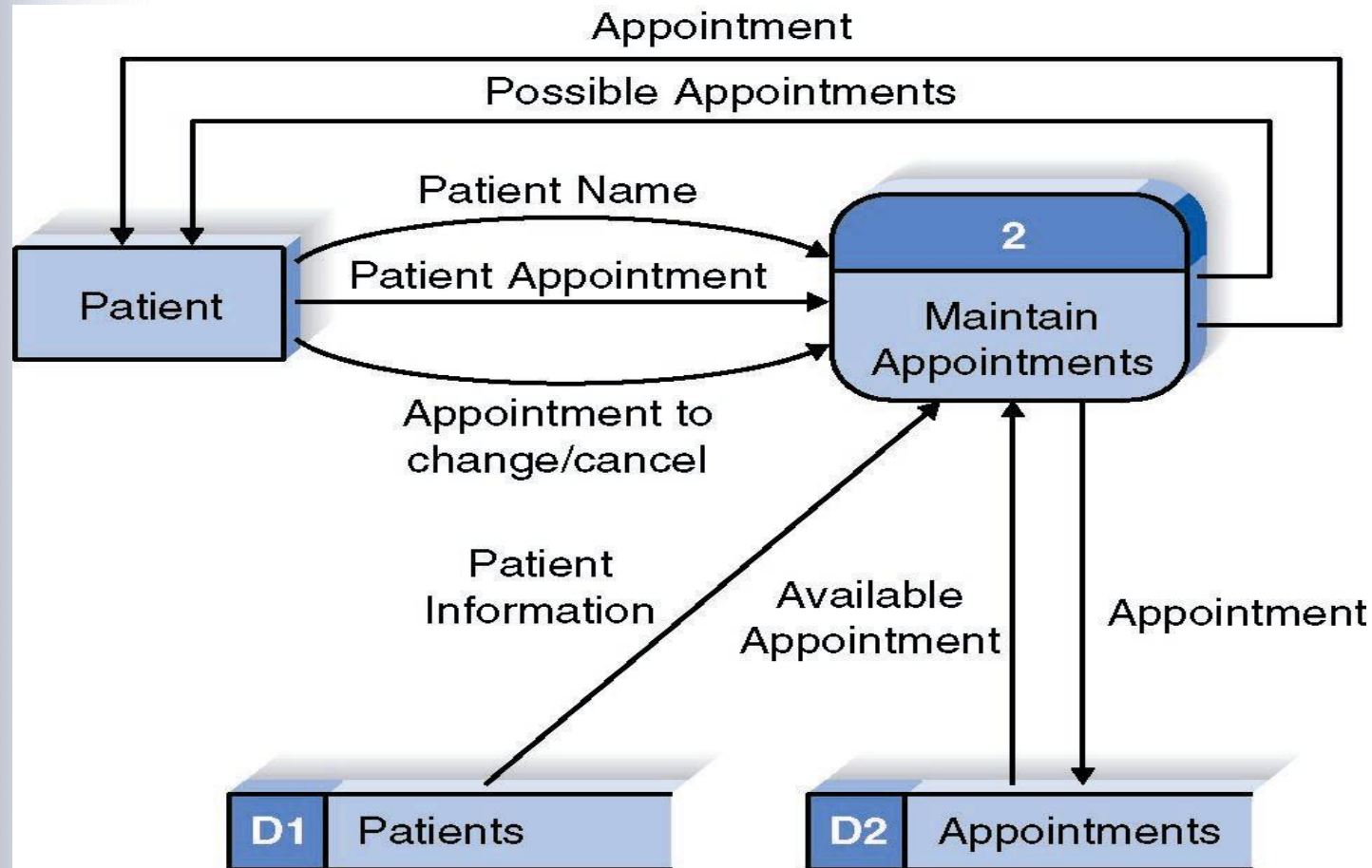
DFD – Common Errors



DFD – Packet Concept



A Second DFD Fragment Example

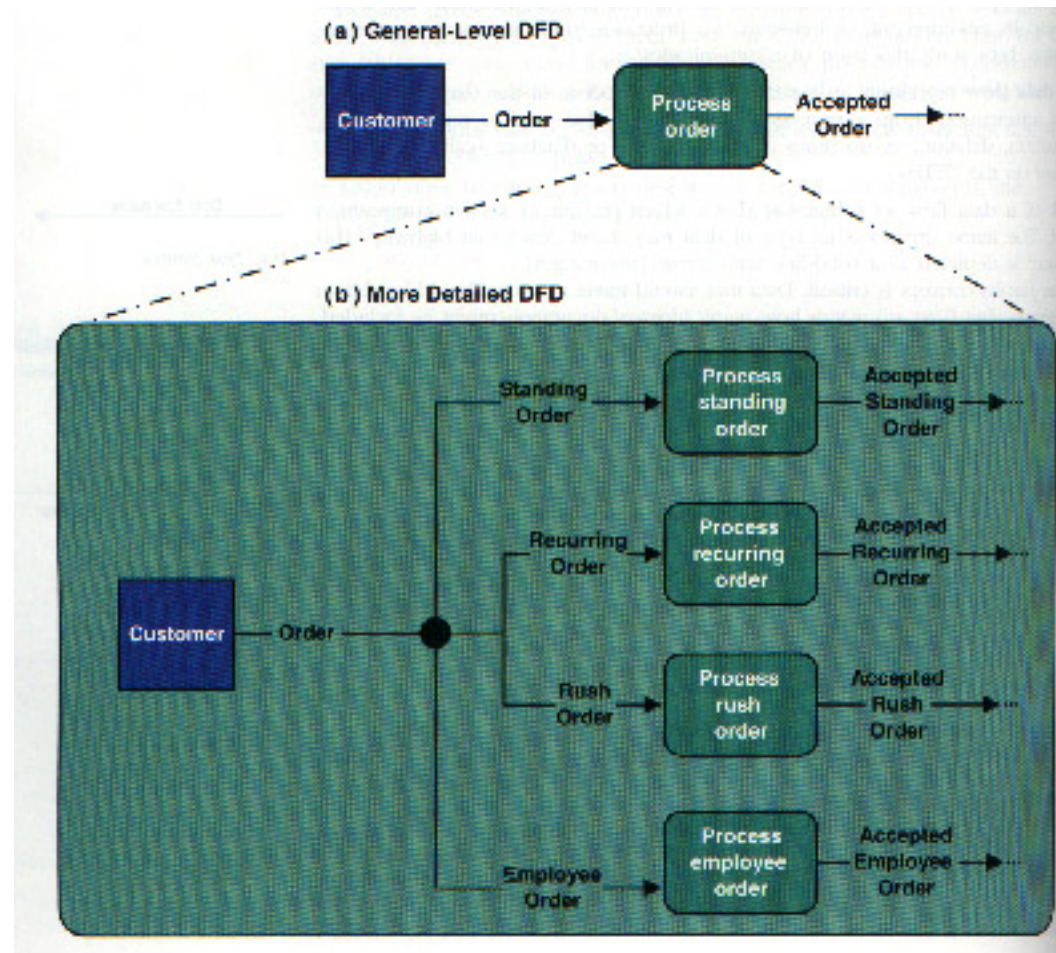


Level 0 Tips



- ☑ Generally move from top to bottom, left to right
- ☑ Minimize crossed lines
- ☑ Iterate as needed
 - ▣ *The DFD is often drawn many times before it is finished, even with very experienced systems analysts*

Composite & Elementary Flows

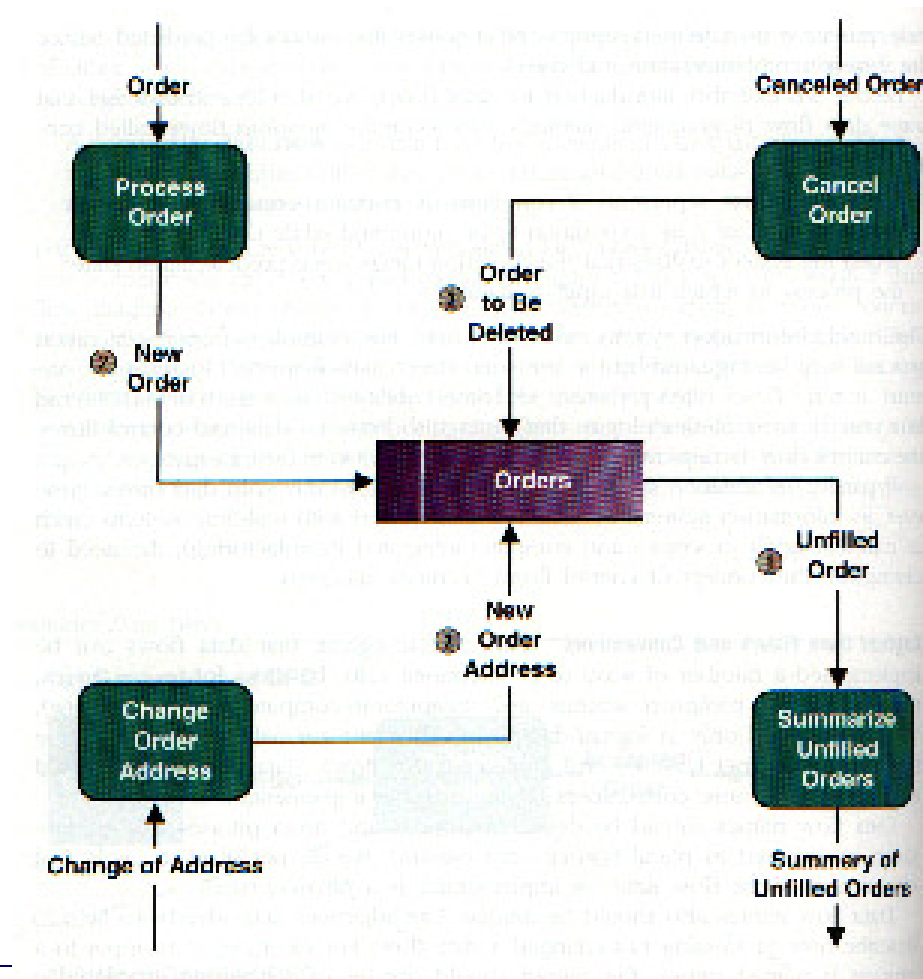


Tips for Level 1 and Below

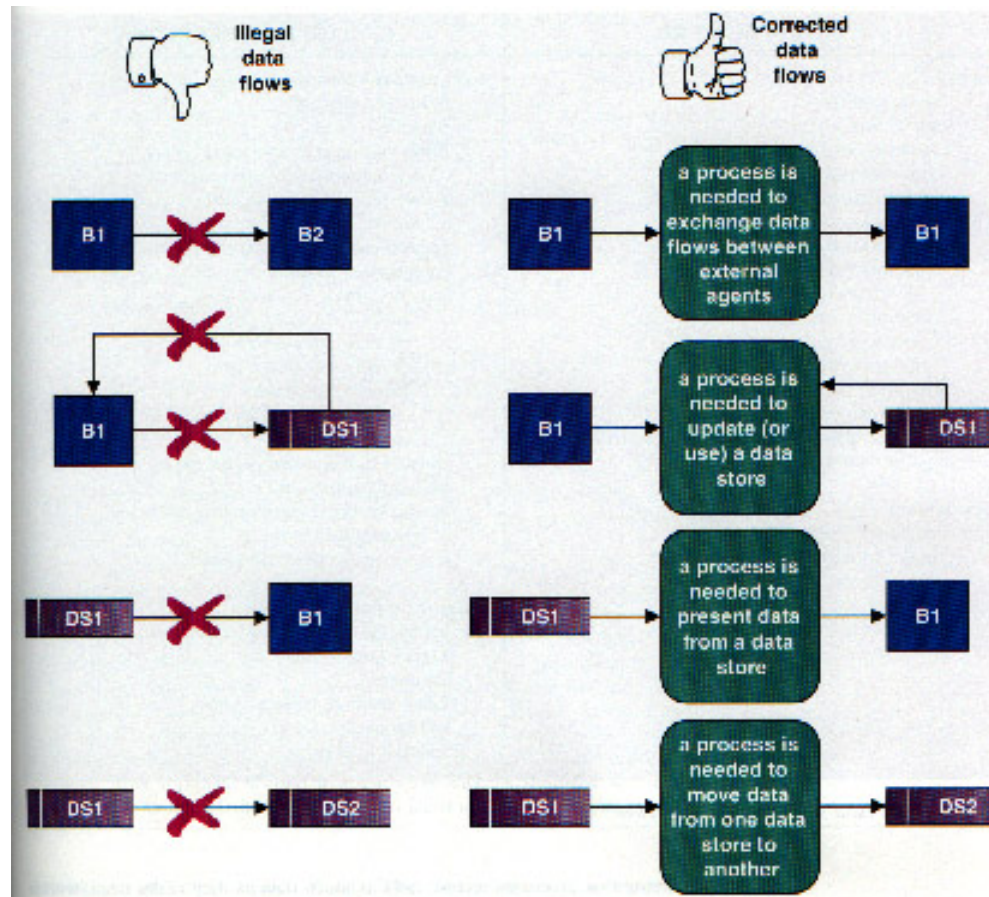


- ☑ Sources for inputs and outputs listed at higher level
- ☑ List source and destination of data flows to processes and stores within each DFD
- ☑ Depth of DFD depends on overall system complexity
 - Two processes generally don't need lower level
 - More than seven processes become overly complex and difficult to read

Flows to & from Data Stores



Illegal Data Flows




Validating the DFD



- ☑ Syntax errors
 - Assure correct DFD structure
- ☑ Semantics errors
 - Assure accuracy of DFD relative to actual/desired business processes
- ☑ User walkthroughs
- ☑ Role-play processes
- ☑ Examine lowest level DFDs
- ☑ Examine names carefully

More on Use Cases, Requirements, Projects



How should we gather requirements? Use cases?



1. Use cases hold functional requirements in an easy-to-read text format
2. They make a good framework for non-functional requirements & project details.
3. Use cases show only the Functional req'ts.
4. Design is not done only in use case units.

A use case collects related scenarios of a user's goal succeeding or failing



UC 4: *Place an order*

Main scenario:

1. User identifies the customer, item and quantity.
2. System accepts and queues the order.

Extensions:

- 1a. Low credit: Customer is 'Preferred' ...
- 1b. Low credit & not Preferred customer: ...
- 2a. Low on stock: Customer accepts reduced...

What sort of a use case fits your project: Fully Dressed, ...?

Use Case 12. *Buy stocks over the web*

Primary Actor: Purchaser (user) **Scope:** PAF

Level: user goal **Precondition:** User already has PAF open.

Guarantees: sufficient log information exists that PAF can detect what went wrong.

Success Guarantees: remote web site acknowledged purchase, user's portfolio updated.

Main success scenario:

1. User selects to buy stocks over the web.
2. PAF gets name of web site to use (E*Trade, Schwabb, etc.)
3. PAF opens web connection to the site, retaining control.
4. User browses and buys stock from the web site.
5. PAF intercepts responses from the web site, and updates the user's portfolio.
6. PAF shows the user the new portfolio standing.

Extensions:

2a. User wants a web site PAF does not support:

2a1. System gets new suggestion from user, with option to cancel use case.


What sort of a use case fits your project: ..., Casual, ...?



☑ Use Case: *Buy something*

- The Requestor initiates a request and sends it to her or his Approver, who completes the request for submission and sends it to the Buyer. The Buyer finds the best vendor, initiates PO with Vendor.
- At any time prior to receiving goods, Requestor can change or cancel the request. Canceling it removes it from any active processing.

What sort of a use case fits your project: ..., or Brief?



Actor	Goal	Brief Description
Production Staff	Prepare digital cartographic source	Convert external digital data to standard format, validate & correct in preparation for merging with operational database.
...

Answer is, of course...

What are your project's

Priorities and Communication Channels?

Use cases do not collect formulae, state, cardinality, performance, uptime, ...



☑ Examples:

1. Order cost = order item costs * 1.06 tax
2. Promotions may not run longer than 6 months.
3. Customers only become Preferred after ...
4. A customer has one and only one sales contact.
5. Response time is ...
6. Uptime requirement is ...
7. Number of simultaneous users will be ...

Capture those in any form available
(*“just a tool problem”*)

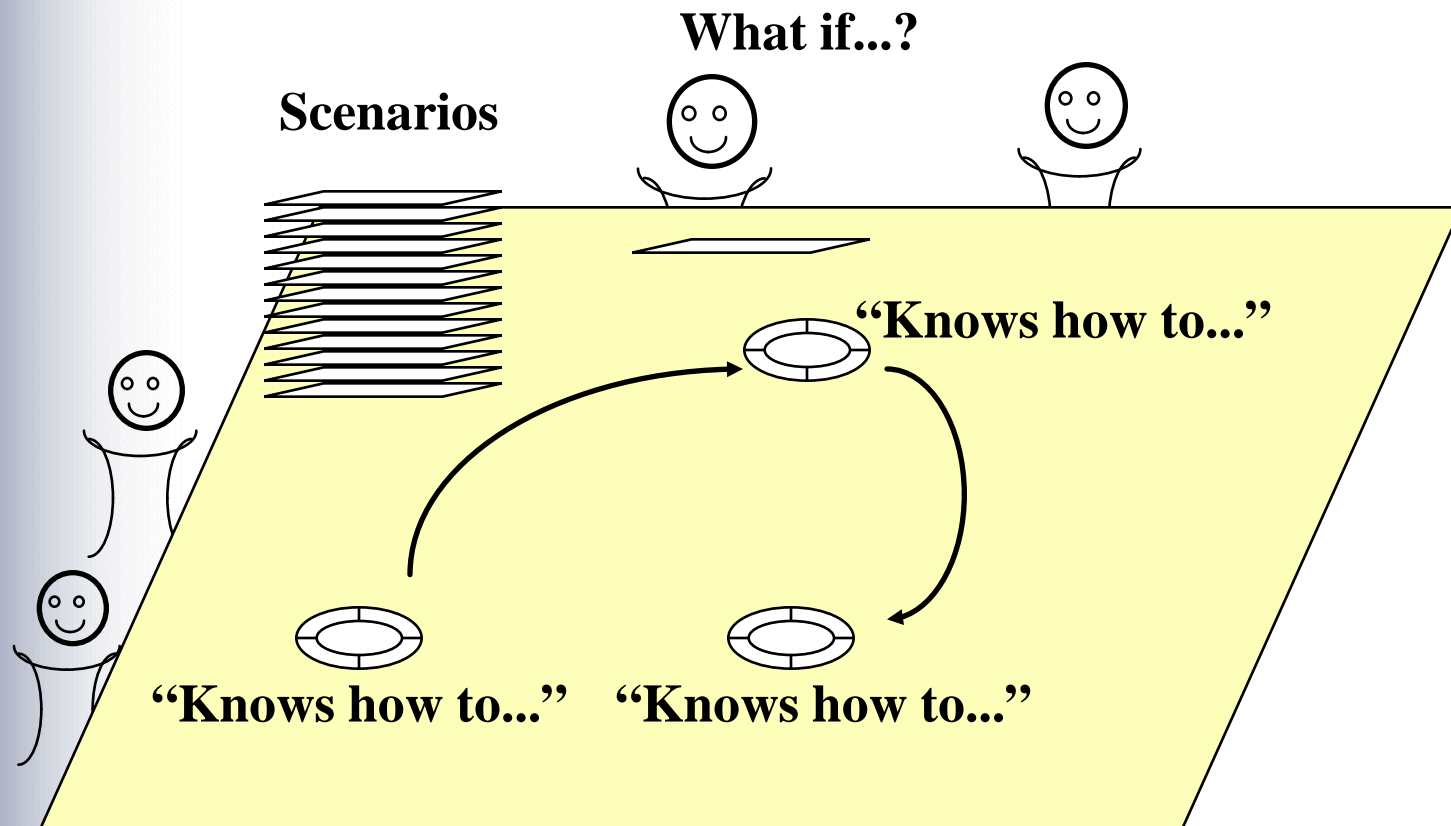
Goals make a good structure on which to hang requirements & project details.



- ☑ Project planning capitalizes on goal structure:
 - Useable Releases.
 - Priorities,
 - Schedule, staffing

Name	P. Actor	Pr.	Diff.	Rel	Tm
Update customer	Customer	high	med	1	..
Generate invoice	Finance	high	high	3	..
Funds transfer	Finance	med	high	4	..
Scan products	Customer	high	high	1	..

Use the scenarios directly to Design and Test the system.



Summary



- ❑ The Data Flow Diagram (DFD) is an essential tool for creating formal descriptions of business processes and data flows.
- ❑ Use cases record the input, transformation, and output of business processes.
- ❑ Eliciting scenario descriptions and modeling business processes are critically important skills for the systems analyst to master.