Process Modeling

Objectives

- Define systems modeling and differentiate logical and physical models.
- Define process modeling and explain its benefits.
- Recognize and understand basic concepts and constructs of a process model.
- Read and interpret a data flow diagram.
- Explain when to construct process models and where to store them.
- Construct a context diagram to illustrate a system’s interfaces with its environment.
- Identify use cases, external and temporal business events.
- Perform event partitioning and organize events in a functional decomposition diagram.
- Draw event diagrams and merge them into a system diagram.
- Draw primitive data flow diagrams and describe the elementary data flows in terms of data structures and procedural logic.
- Document the distribution of processes to locations.
- Synchronize data and process models using a CRUD matrix.

Models: Logical and Physical

Model – a pictorial representation of reality.

Just as a picture is worth a thousand words, most models are pictorial representations of reality.

Logical model – a nontechnical pictorial representation that depicts what a system is or does. Synonyms or *essential model*, *conceptual model*, and *business model*.

Physical model – a technical pictorial representation that depicts what a system is or does and how the system is implemented. Synonyms are *implementation model* and *technical model*.

Why Logical System Models

- Logical models remove biases that are the result of the way the system is currently implemented, or the way that any one person thinks the system might be implemented.
- Logical models reduce the risk of missing business requirements because we are too preoccupied with technical results.
- Logical models allow us to communicate with end-users in nontechnical or less technical languages.

Process Modeling and DFDs

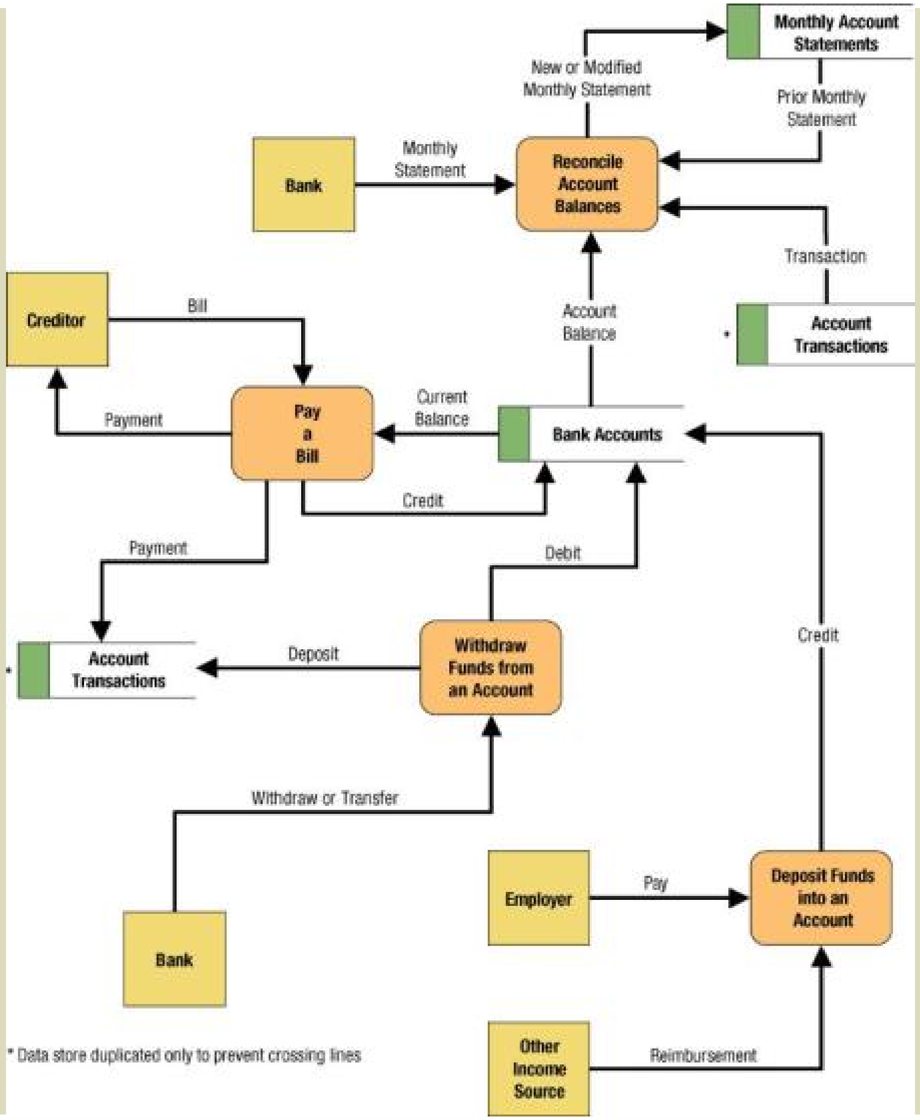
Process modeling – a technique used to organize and document a system's processes.

- Flow of data through processes
- Logic
- Policies
- Procedures

Data flow diagram (DFD) – a process model used to depict the flow of data through a system and the work or processing performed by the system. Synonyms are bubble chart, transformation graph, and process model.

- The DFD has also become a popular tool for business process redesign.

Simple Data Flow Diagram



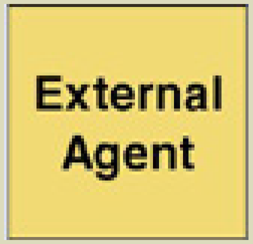
Differences Between DFDs and Flowcharts

- Processes on DFDs can operate in parallel (at-the-same-time)
  - Processes on flowcharts execute one at a time
- DFDs show the flow of data through a system
  - Flowcharts show the flow of control (sequence and transfer of control)
- Processes on a DFD can have dramatically different timing (daily, weekly, on demand)
  - Processes on flowcharts are part of a single program with consistent timing

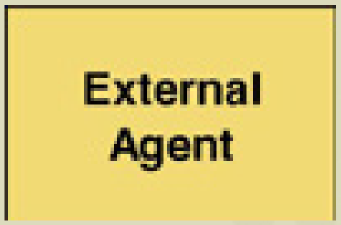
External Agents

External agent – an outside person, unit, system, or organization that interacts with a system. Also called an *external entity*.

- External agents define the “boundary” or scope of a system being modeled.
- As scope changes, external agents can become processes, and vice versa.
- Almost always one of the following:
  - Office, department, division.
  - An external organization or agency.
  - Another business or another information system.
  - One of system’s end-users or managers
- Named with descriptive, singular noun



Gane and Sarson shape

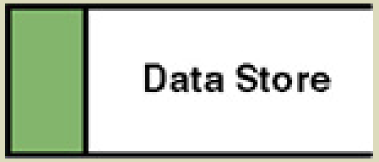


DeMarco/Yourdon shape

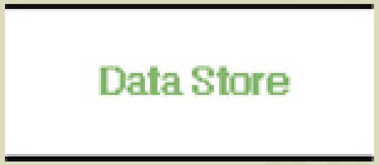
Data Stores

Data store – stored data intended for later use. Synonyms are *file* and *database*.

- Frequently implemented as a file or database.
- A data store is “data at rest” compared to a data flow that is “data in motion.”
- Almost always one of the following:
  - Persons (or groups of persons)
  - Places
  - Objects
  - Events (about which data is captured)
  - Concepts (about which data is important)
- Data stores depicted on a DFD store all instances of data entities (depicted on an ERD)
- Named with plural noun



Gane and Sarson shape

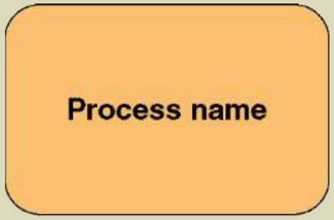


DeMarco/Yourdon shape

Process Concepts

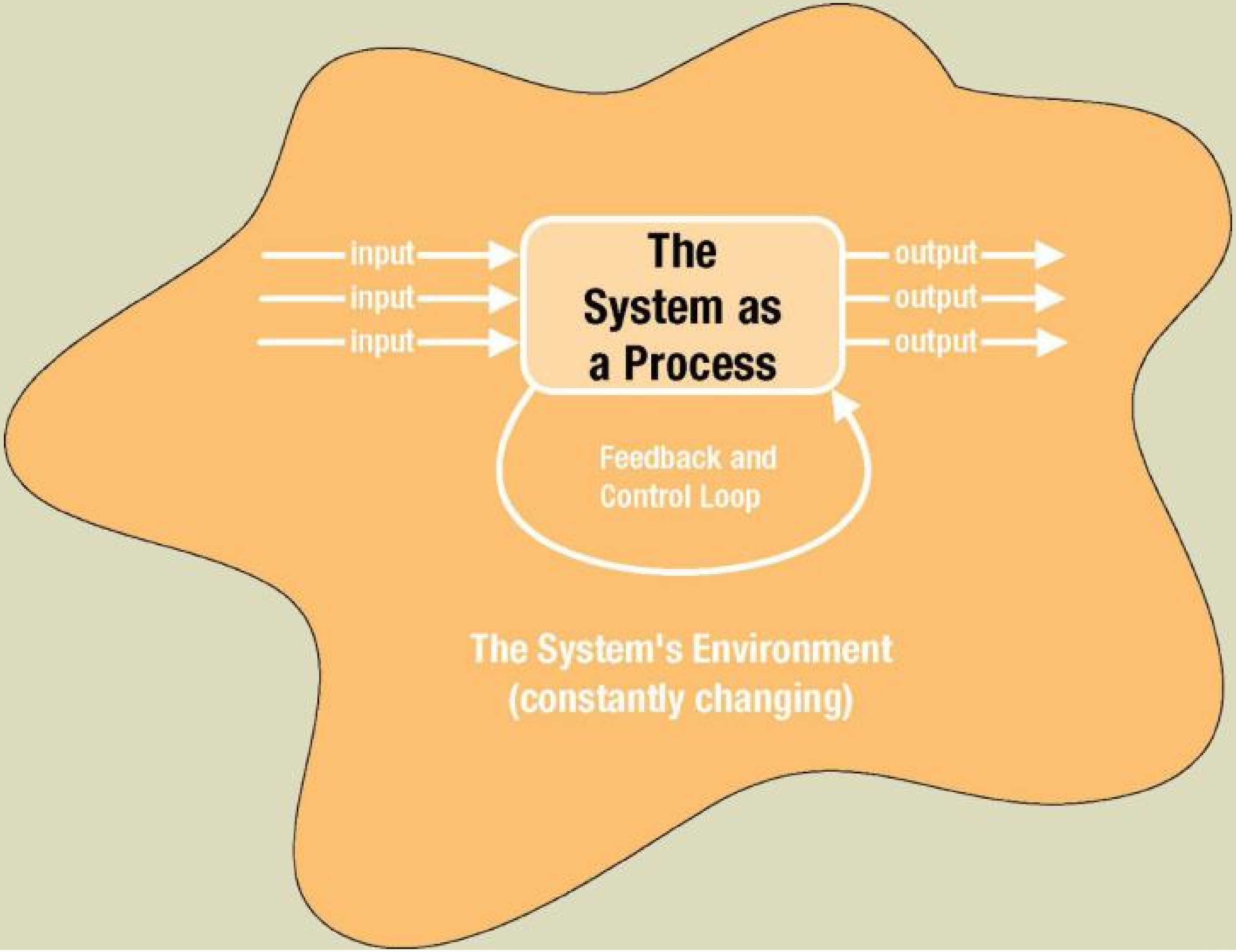
Process – work performed by a system in response to incoming data flows or conditions. A synonym is *transform*.

- All information systems include processes - usually many of them
- Processes respond to business events and conditions and transform data into useful information
- Modeling processes helps us to understand the interactions with the system's environment, other systems, and other processes.
- Named with a strong action verb followed by object clause describing what the work is performed on/for .



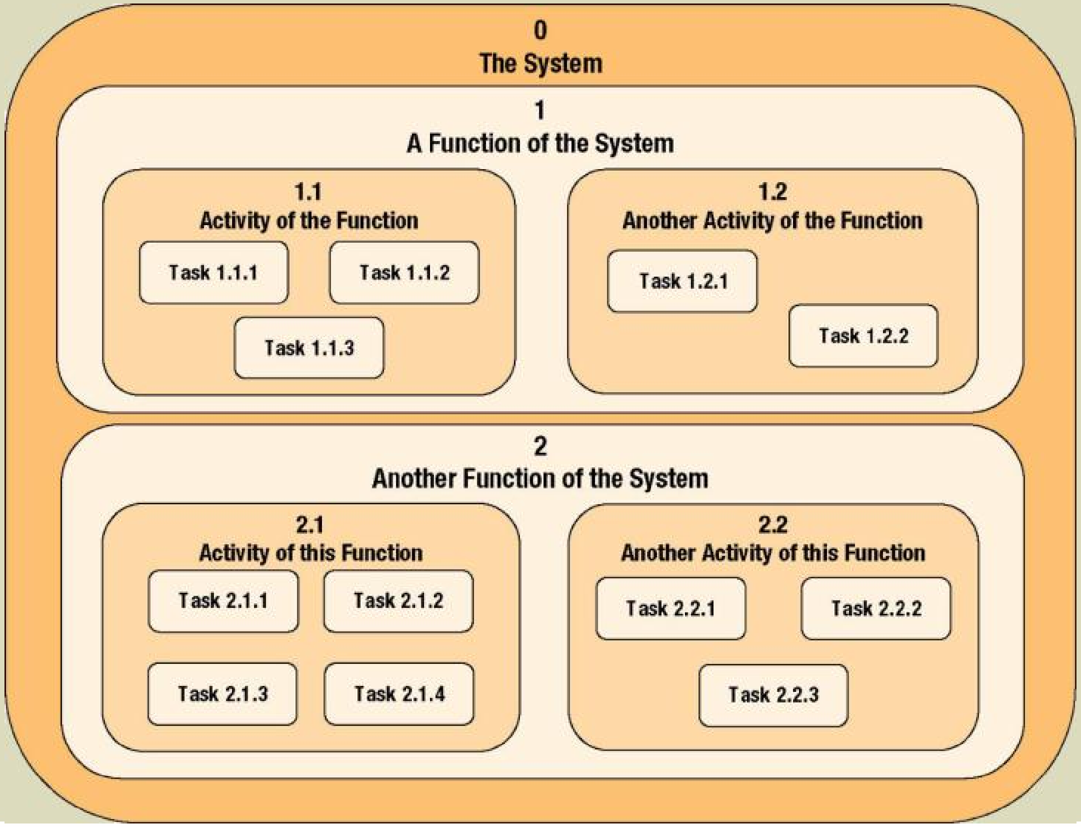
Gane and Sarson shape

The System is Itself a Process


input
input
input
The System as a Process
output
output
output
Feedback and Control Loop
The System's Environment (constantly changing)

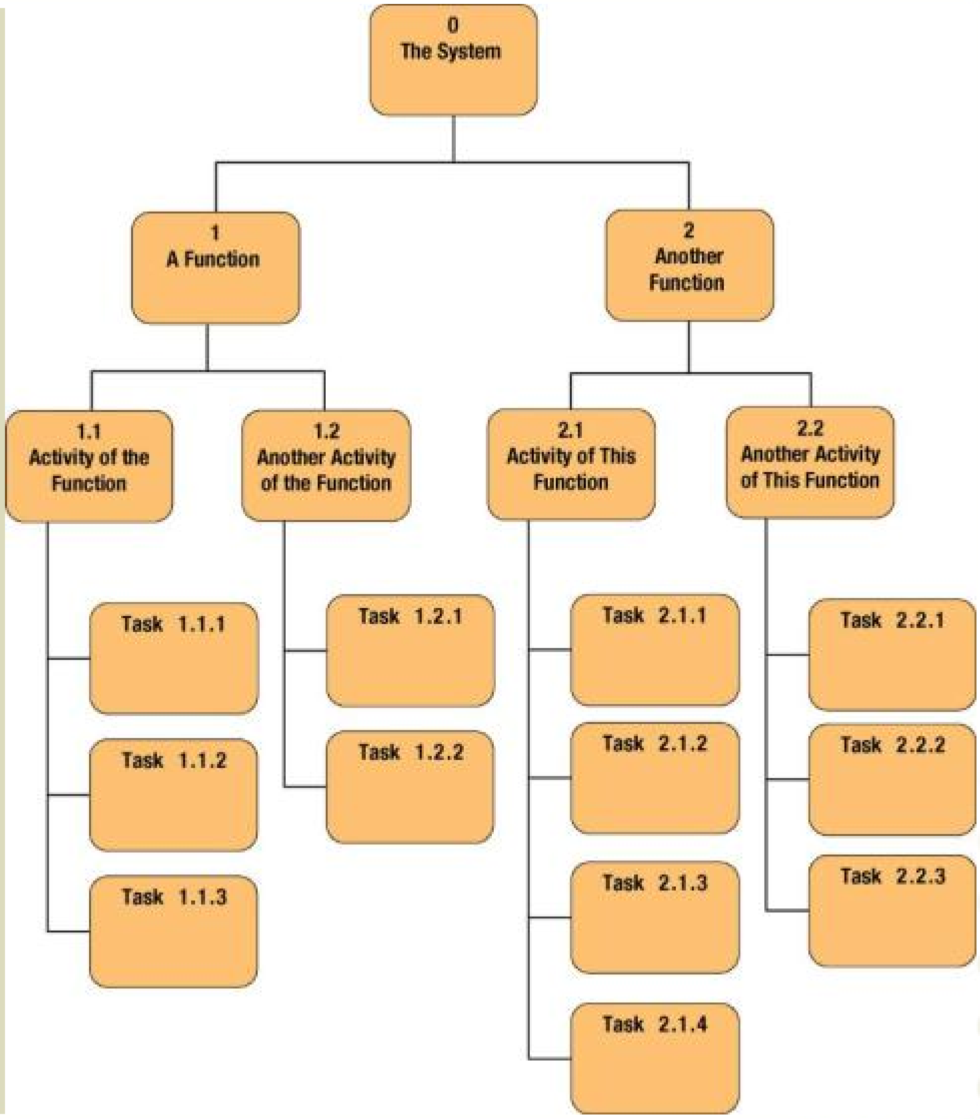
Process Decomposition

Decomposition – the act of breaking a system into sub-components. Each level of abstraction reveals more or less detail.



Decomposition Diagrams

Decomposition diagram – a tool used to depict the decomposition of a system. Also called hierarchy chart.



Types of Logical Processes

Function – a set of related and ongoing activities of a business.

- A function has no start or end.

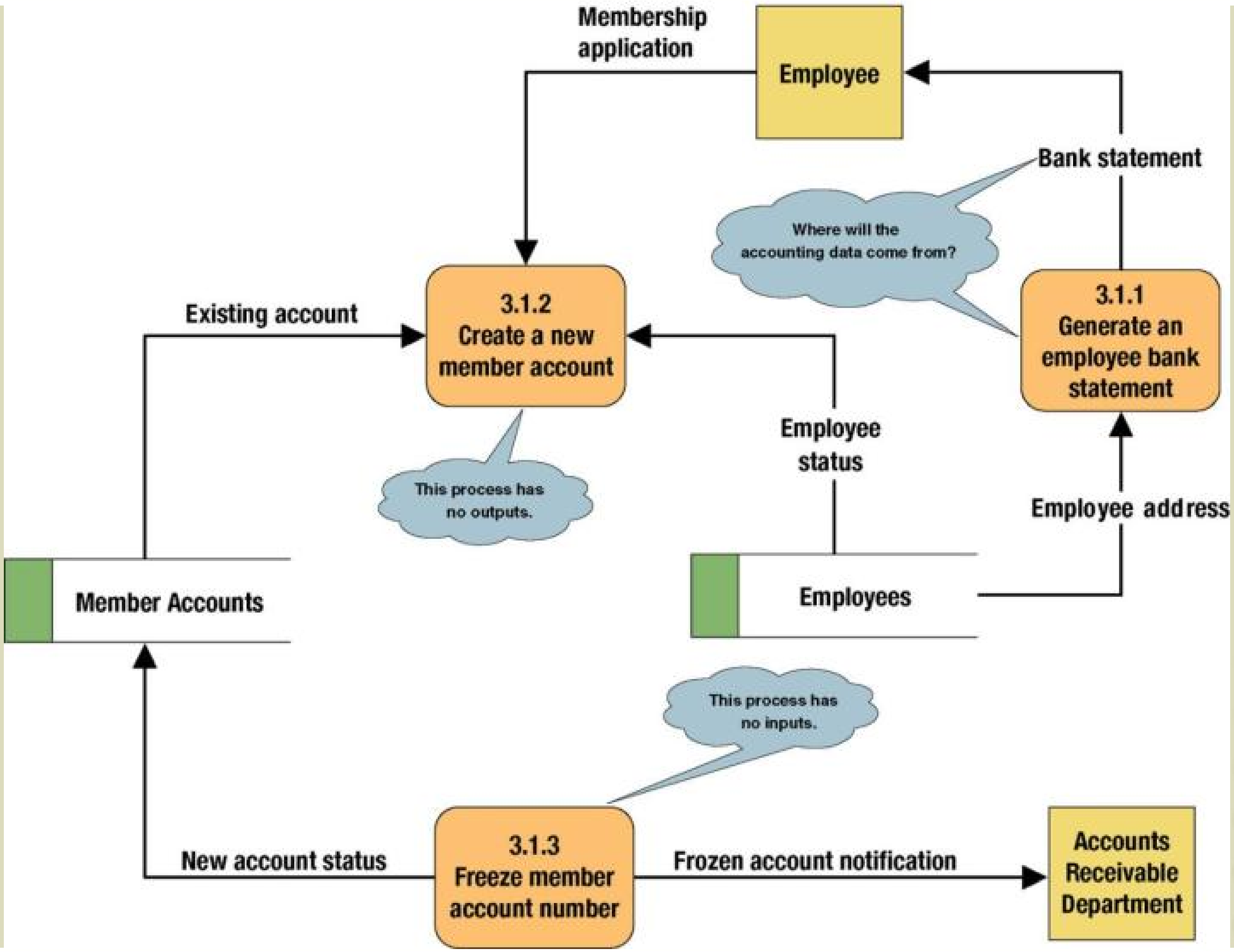
Event – a logical unit of work that must be completed as a whole. Sometimes called a *transaction*.

- Triggered by a discrete input and is completed when process has responded with appropriate outputs.
- Functions consist of processes that respond to events.

Elementary process – a discrete, detailed activity or task required to complete the response to an event. Also called a *primitive process*.

- The lowest level of detail depicted in a process model.

Common Process Errors on DFDs

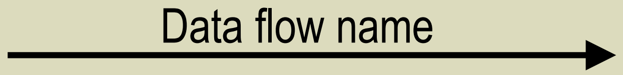


Membership application
Employee
Bank statement
Where will the accounting data come from?
Existing account
3.1.2 Create a new member account
3.1.1 Generate an employee bank statement
Employee status
This process has no outputs.
Employee address
Member Accounts
Employees
This process has no inputs.
New account status
3.1.3 Freeze member account number
Frozen account notification
Accounts Receivable Department

Data Flows & Control Flows

Data flow – data that is input to or output from a process.

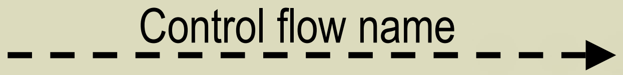
- A data flow is data in motion
- A data flow may also be used to represent the creation, reading, deletion, or updating of data in a file or database (called a data store).



Composite data flow – a data flow that consists of other data flows.

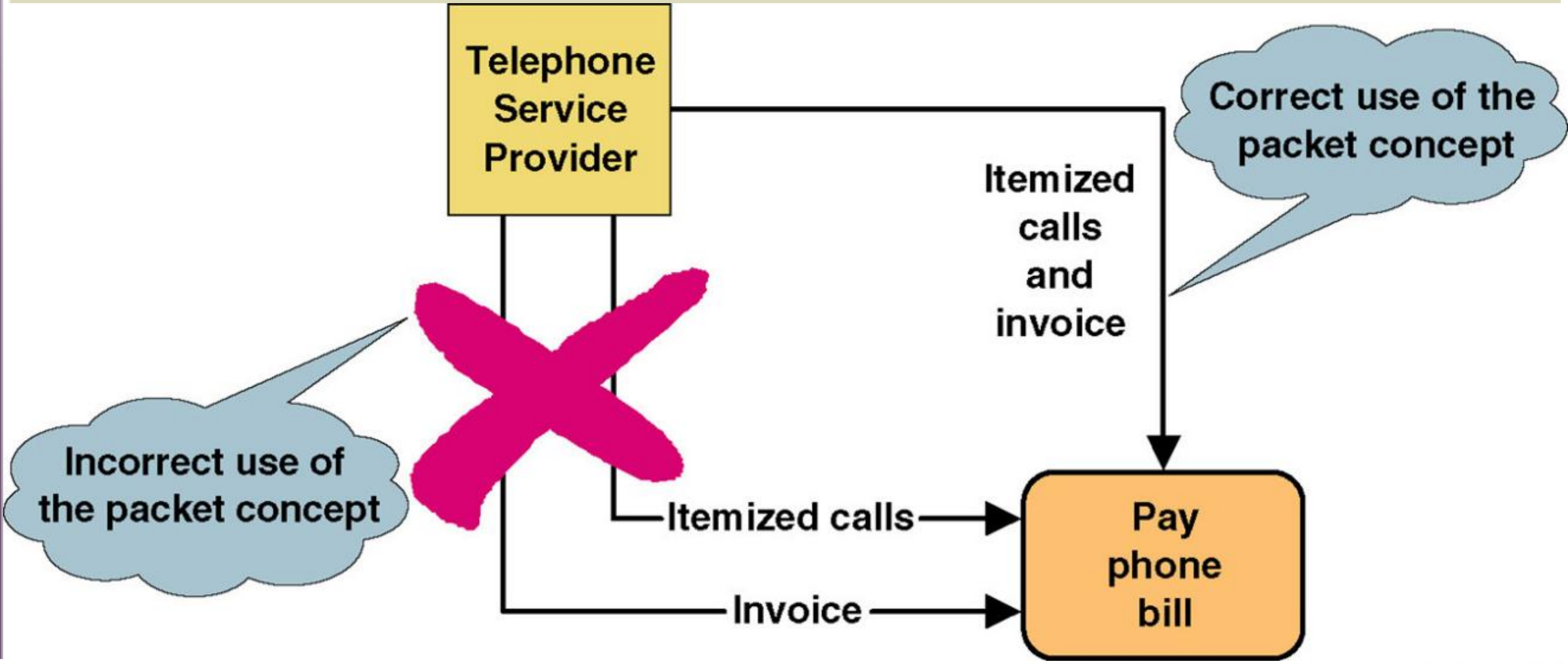
Control flow – a condition or nondata event that triggers a process.

- Used sparingly on DFDs.

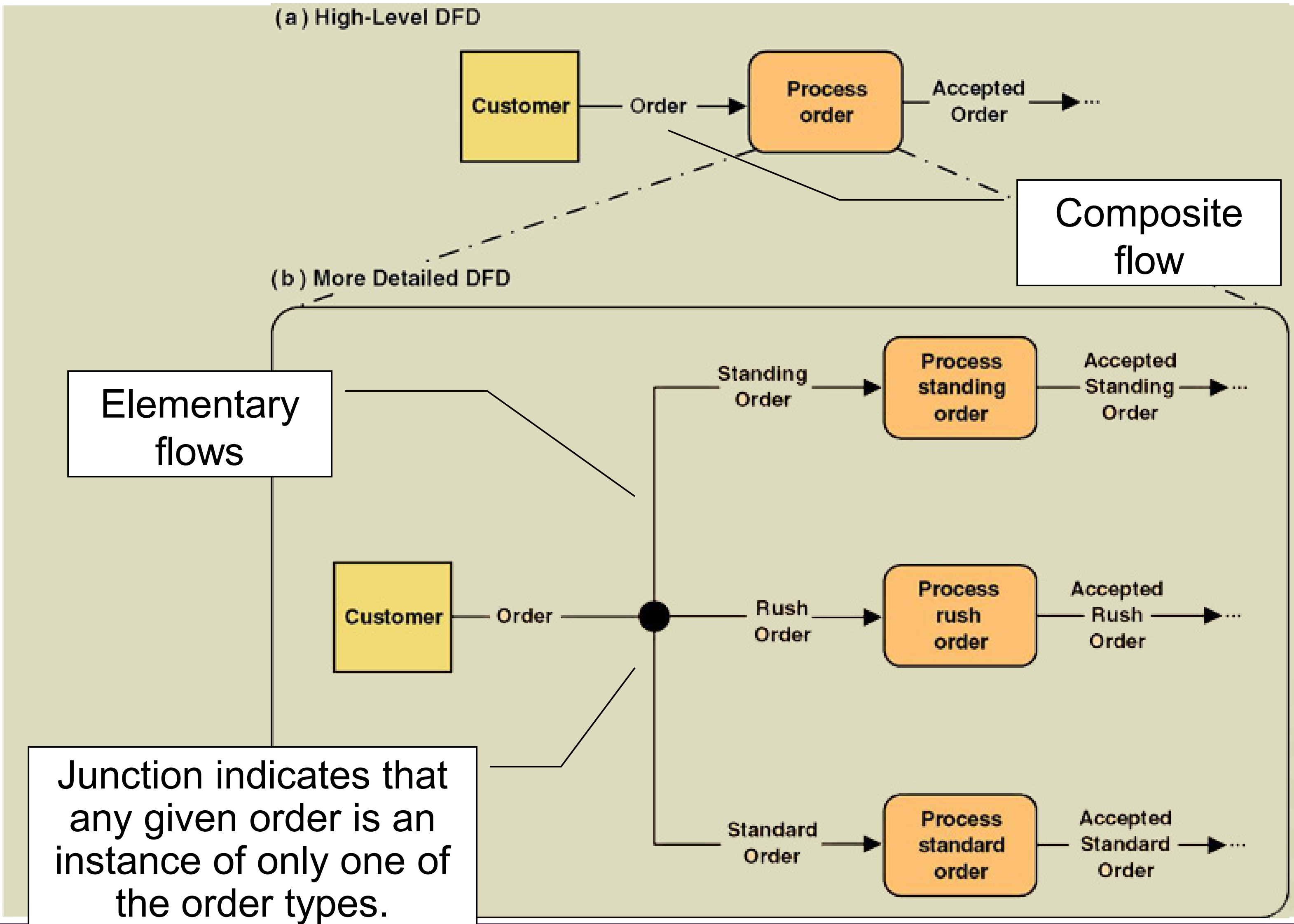


Data Flow Packet Concept

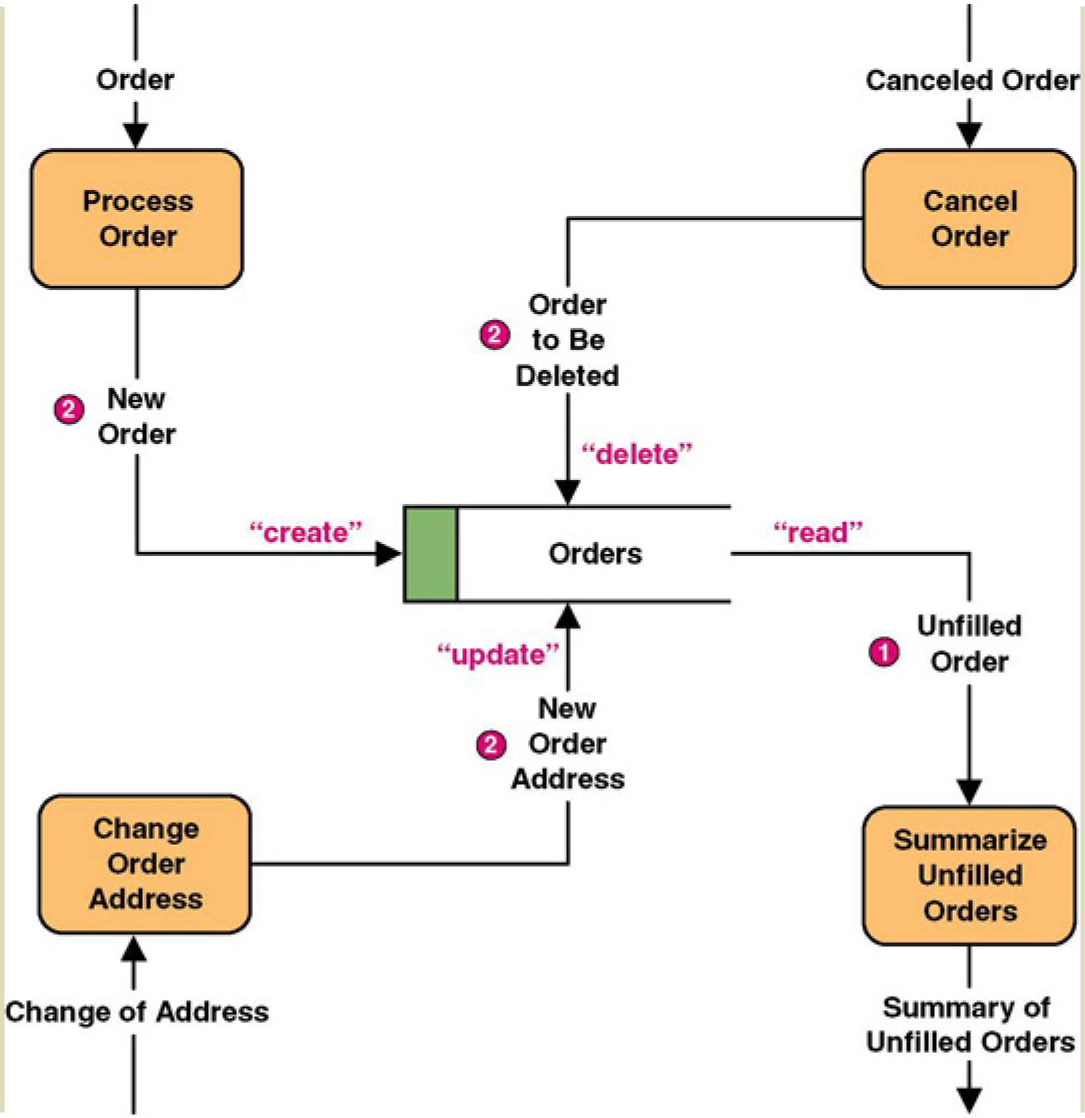
- Data that should travel together should be shown as a single data flow, no matter how many physical documents might be included.


Telephone Service Provider
Correct use of the packet concept
Itemized calls and invoice
Incorrect use of the packet concept
Itemized calls
Invoice
Pay phone bill

Composite and Elementary Data Flows

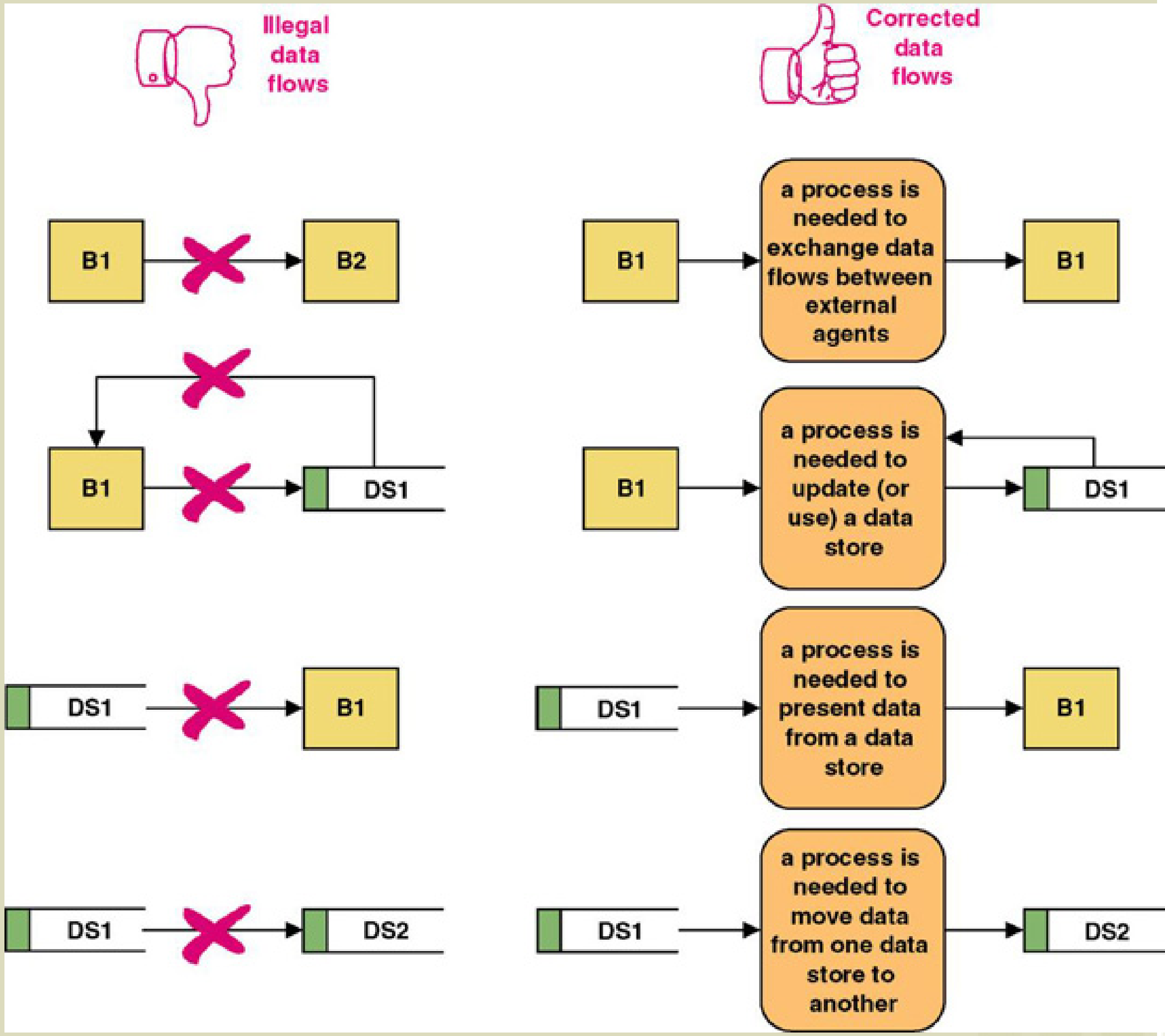


Data Flows to and from Data Stores



Rules for Data Flows

- A data flow should never go unnamed.
- In logical modeling, data flow names should describe the data flow without describing the implementation
- All data flows must begin and/or end at a process.


Illegal data flows
Corrected data flows
B1
B2
B1
DS1
DS1
B1
DS1
DS2
B1
a process is needed to exchange data flows between external agents
B1
B1
a process is needed to update (or use) a data store
DS1
DS1
a process is needed to present data from a data store
B1
DS1
a process is needed to move data from one data store to another
DS2

Data Conservation

Data conservation – the practice of ensuring that a data flow contains only data needed by the receiving process.

- Sometimes called *starving the processes*.
- New emphasis on business process redesign to identify and eliminate inefficiencies.
- Simplifies the interface between those processes.
- Must precisely define the data composition of each data flow, expressed in the form of *data structures*.

Data Types and Domains

Data attributes should be defined by data types and domains.

Data type - a class of data that be stored in an attribute.

- Character, integers, real numbers, dates, pictures, etc.

Domain – the legitimate values for an attribute.

Diverging and Converging Data Flows

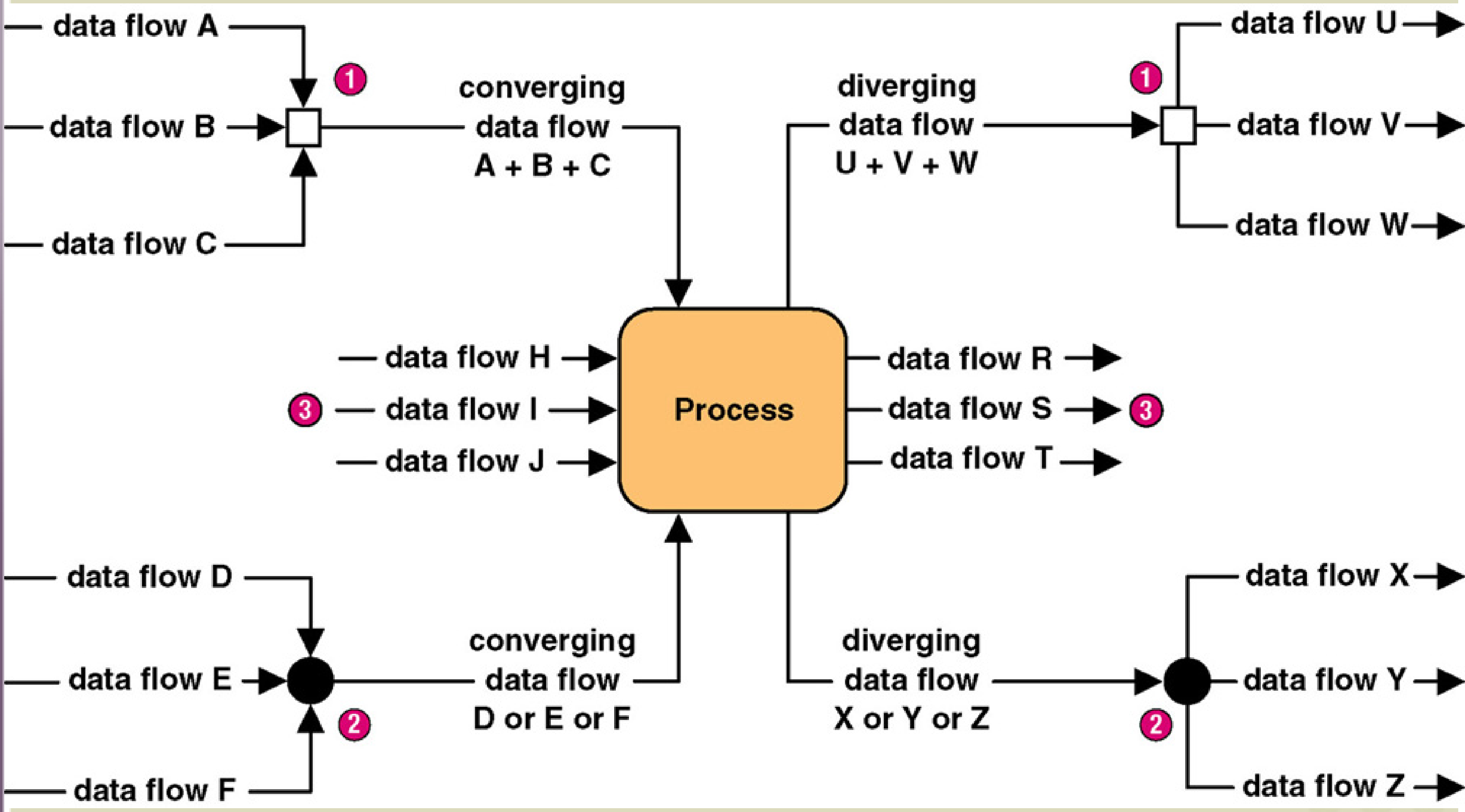
Diverging data flow – a data flow that splits into multiple data flows.

- Indicates data that starts out naturally as one flow, but is routed to different destinations.
- Also useful to indicate multiple copies of the same output going to different destinations.

Converging data flow – the merger of multiple data flows into a single packet.

- Indicates data from multiple sources that can (must) come together as a single packet for subsequent processing.

Diverging and Converging Data Flows



When to Draw Process Models

- Strategic systems planning
  - Enterprise process models illustrate important business functions.
- Business process redesign
  - “As is” process models facilitate critical analysis.
  - “To be” process models facilitate improvement.
- Systems analysis (primary focus of this course)
  - Model existing system including its limitations
  - Model target system’s logical requirements
  - Model candidate technical solutions
  - Model the target technical solution

Classical Structured Analysis

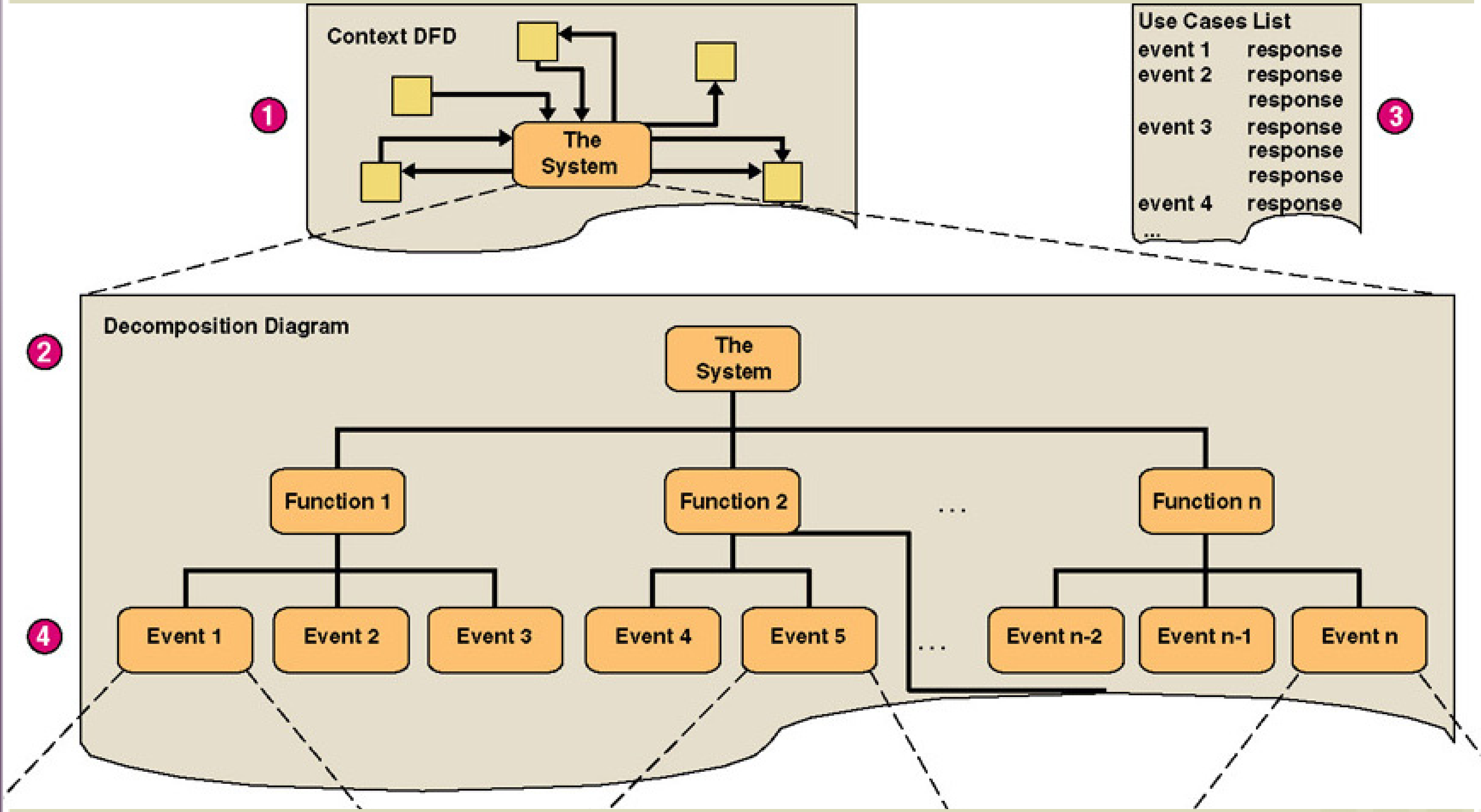
Rarely practiced anymore because cumbersome & time-consuming

1. Draw top-down physical DFDs that represent current physical implementation of the system.
2. Convert physical DFDs to logical equivalents.
3. Draw top-down logical DFDs that represent improved system.
4. Describe all data flows, data stores, policies, and procedures in data dictionary or encyclopedia.
5. Optionally, mark up copies of the logical DFDs to represent alternative physical solutions.
6. Draw top-down physical DFDs representing target solution.

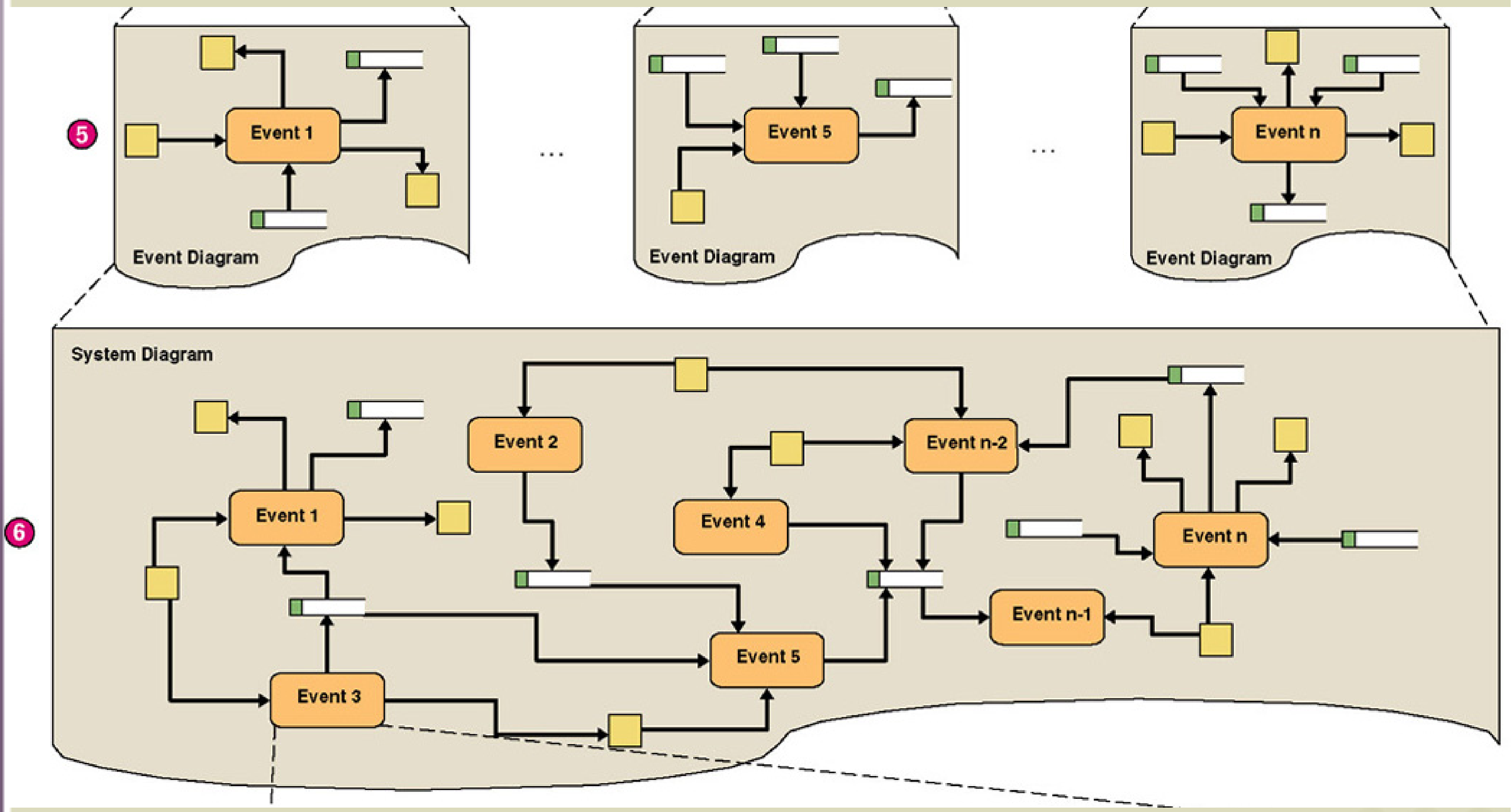
Modern Structured Analysis (More Commonly Practiced)

1. Draw context DFD to establish initial project scope.
2. Draw functional decomposition diagram to partition the system into subsystems.
3. Create event-response or use-case list for the system to define events for which the system must have a response.
4. Draw an event DFD (or event handler) for each event.
5. Merge event DFDs into a system diagram (or, for larger systems, subsystem diagrams).
6. Draw detailed, primitive DFDs for the more complex event handlers.
7. Document data flows and processes in data dictionary.

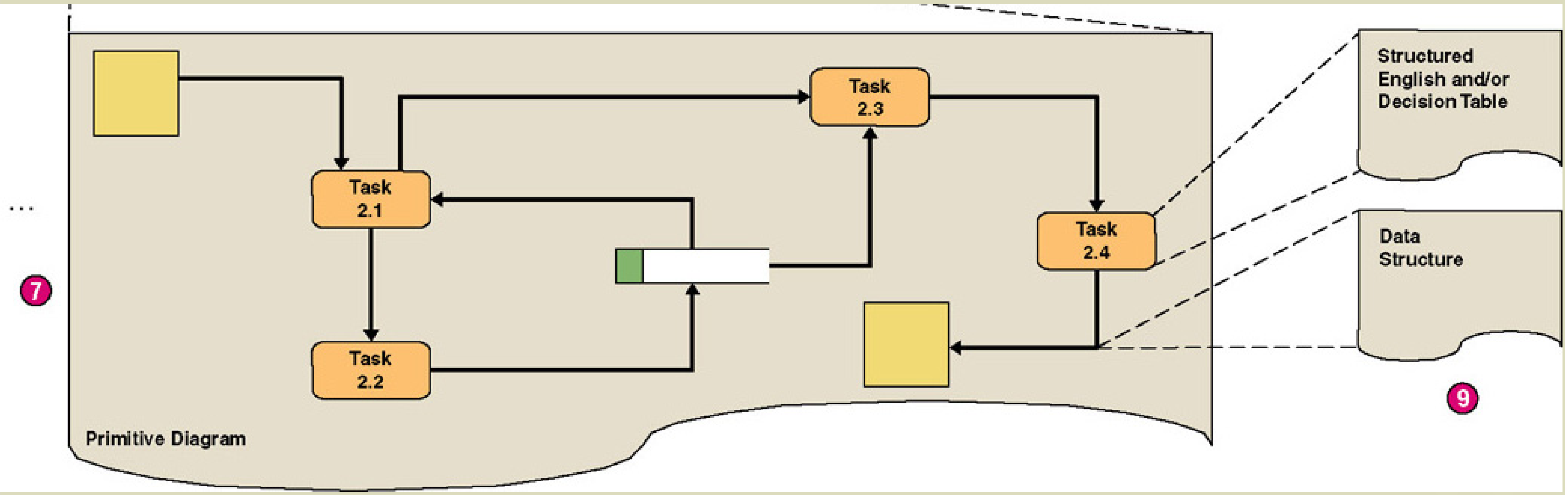
Structured Analysis Diagram Progression (1 of 3)



Structured Analysis Diagram Progression (2 of 3)



Structured Analysis Diagram Progression (3 of 3)

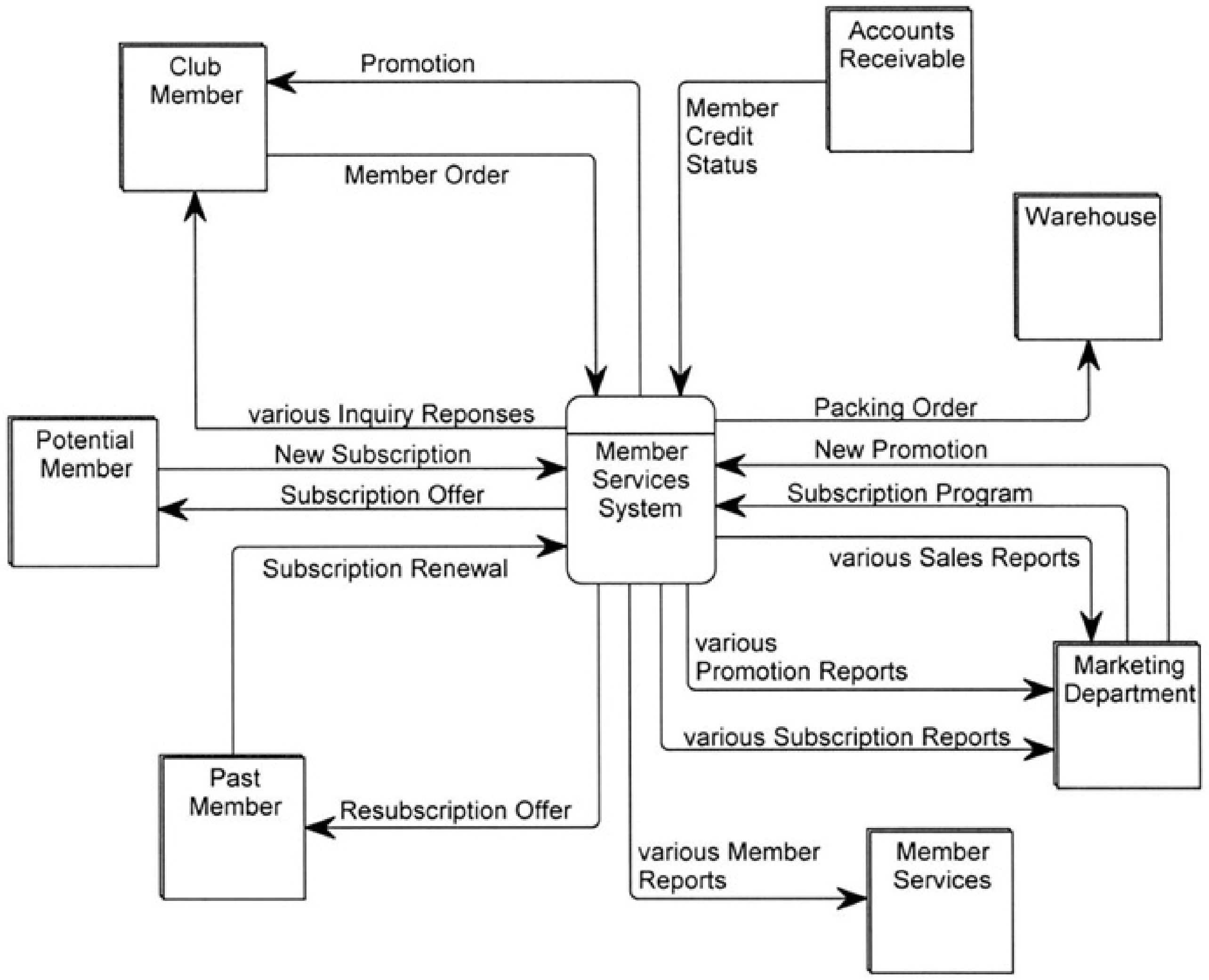


Context Data Flow Diagram

- **Context data flow diagram** - a process model used to document the scope for a system. Also called the environmental model.

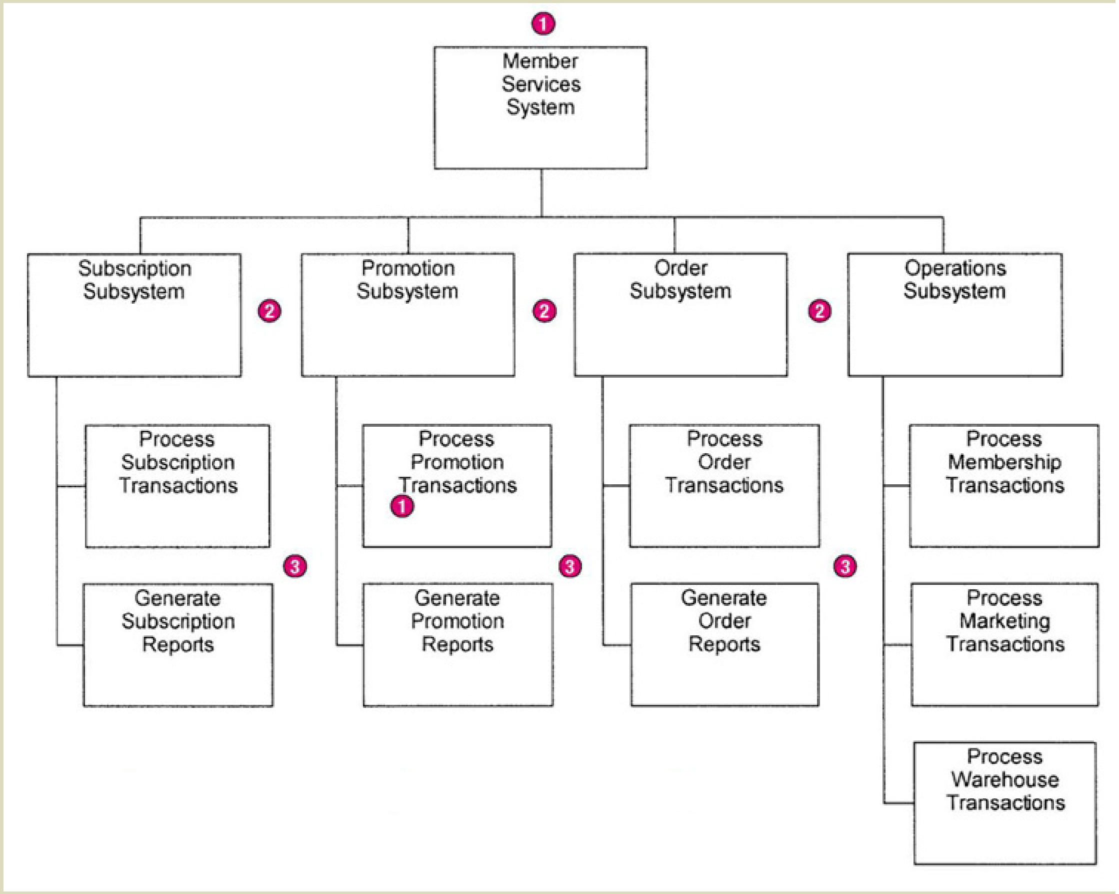
1. Think of the system as a "black box."
2. Ask users what business transactions the system must respond to. These are inputs, and the sources are external agents.
3. Ask users what responses must be produced by the system. These are outputs, and the destinations are external agents.
4. Identify any external data stores, if any.
5. Draw a context diagram.

SoundStage Context DFD



SoundStage Functional Decomposition Diagram

- Break system into sub-components to reveal more detail.
- Every process to be factored should be factored into at least two child processes.
- Larger systems might be factored into subsystems and functions.


1
Member Services System
Subscription Subsystem
2
Promotion Subsystem
2
Order Subsystem
2
Operations Subsystem
Process Subscription Transactions
Process Promotion Transactions
1
Process Order Transactions
Process Membership Transactions
3
3
3
Generate Subscription Reports
Generate Promotion Reports
Generate Order Reports
Process Marketing Transactions
Process Warehouse Transactions

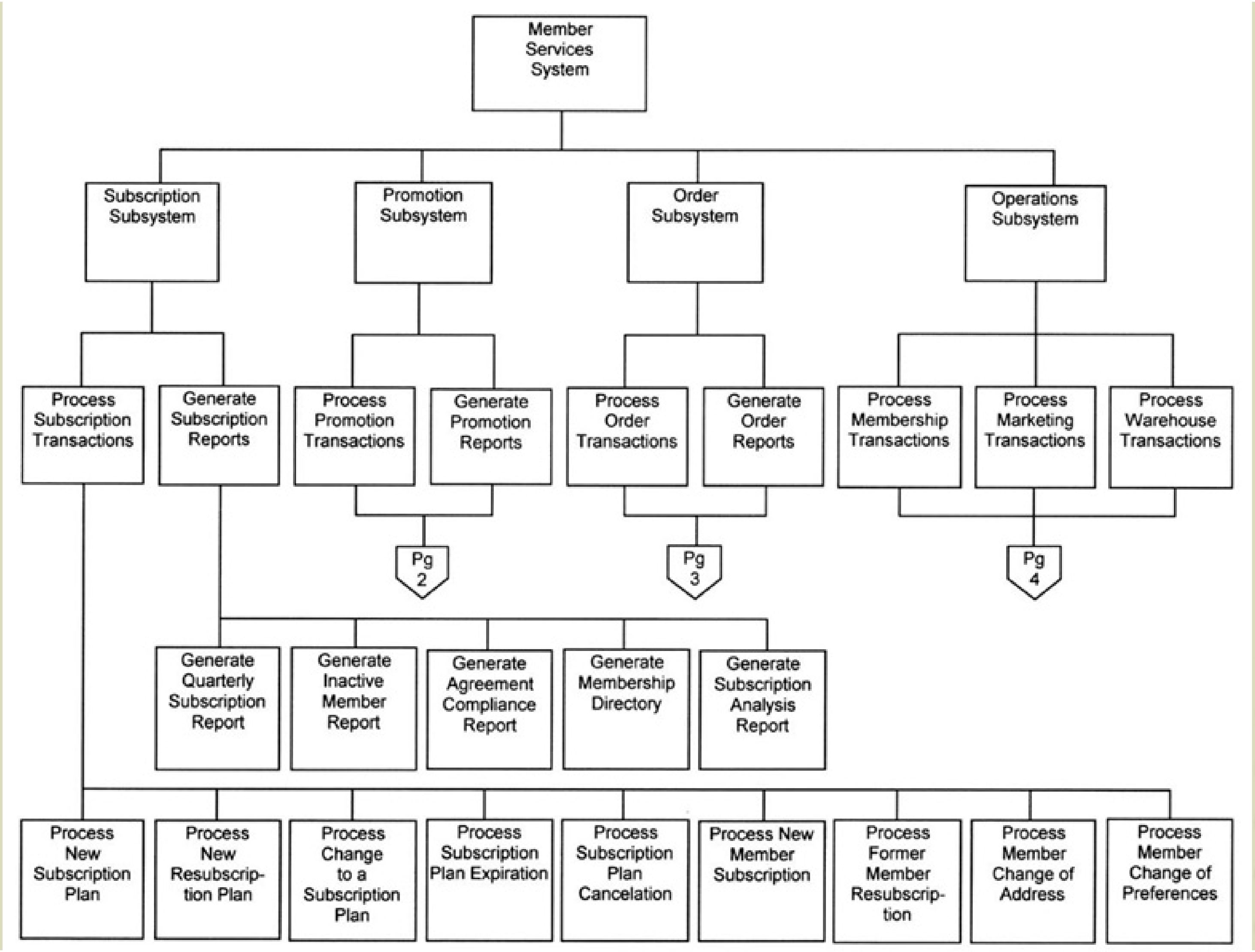
Events and Use Cases

- **External events** are initiated by external agents. They result in an input transaction or data flow.
- **Temporal events** are triggered on the basis of time, or something that merely happens. They are indicated by a control flow.
- **State events** trigger processes based on a system’s change from one state or condition to another. They are indicated by a control flow.
- **Use case** – an analysis tool for finding and identifying business events and responses.
- **Actor** – anything that interacts with a system.

SoundStage Partial Use Case List

Actor/ External Agent	Event (or Use Case)	Trigger	Response
Marketing	Establishes a new membership subscription plan to entice new members.	New Member Subscription Program	Generate Subscription Plan Confirmation. Create Agreement in the database.
Marketing	Establishes a new membership resubscription plan to lure back former members.	Past Member Resubscription Program	Generate Subscription Plan Confirmation. Create Agreement in the database.
(time)	A subscription plan expires.	(current date)	Generate Agreement Change Confirmation. Logically delete Agreement in database.
Member	Joins club by subscribing.	New Subscription	Generate Member Directory Update Confirmation. Create Member in database. Create first Member Order and Member Ordered Products in database.

SoundStage Partial Event Decomposition Diagram

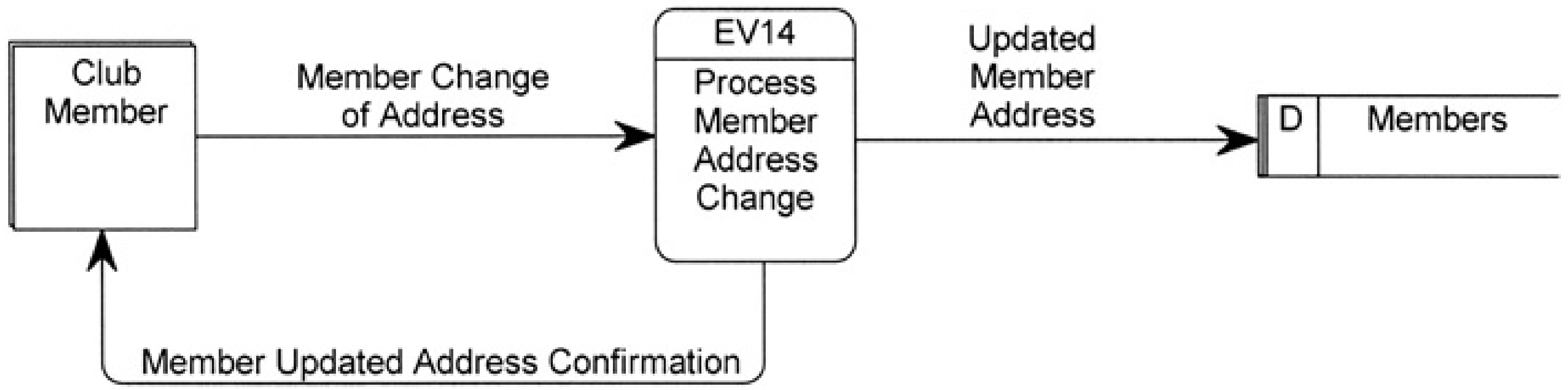

Member Services System
Subscription Subsystem
Promotion Subsystem
Order Subsystem
Operations Subsystem
Process Subscription Transactions
Generate Subscription Reports
Process Promotion Transactions
Generate Promotion Reports
Process Order Transactions
Generate Order Reports
Process Membership Transactions
Process Marketing Transactions
Process Warehouse Transactions
Pg 2
Pg 3
Pg 4
Generate Quarterly Subscription Report
Generate Inactive Member Report
Generate Agreement Compliance Report
Generate Membership Directory
Generate Subscription Analysis Report
Process New Subscription Plan
Process New Resubscrip-tion Plan
Process Change to a Subscription Plan
Process Subscription Plan Expiration
Process Subscription Plan Cancelation
Process New Member Subscription
Process Former Member Resubscrip-tion
Process Member Change of Address
Process Member Change of Preferences

Event Diagrams

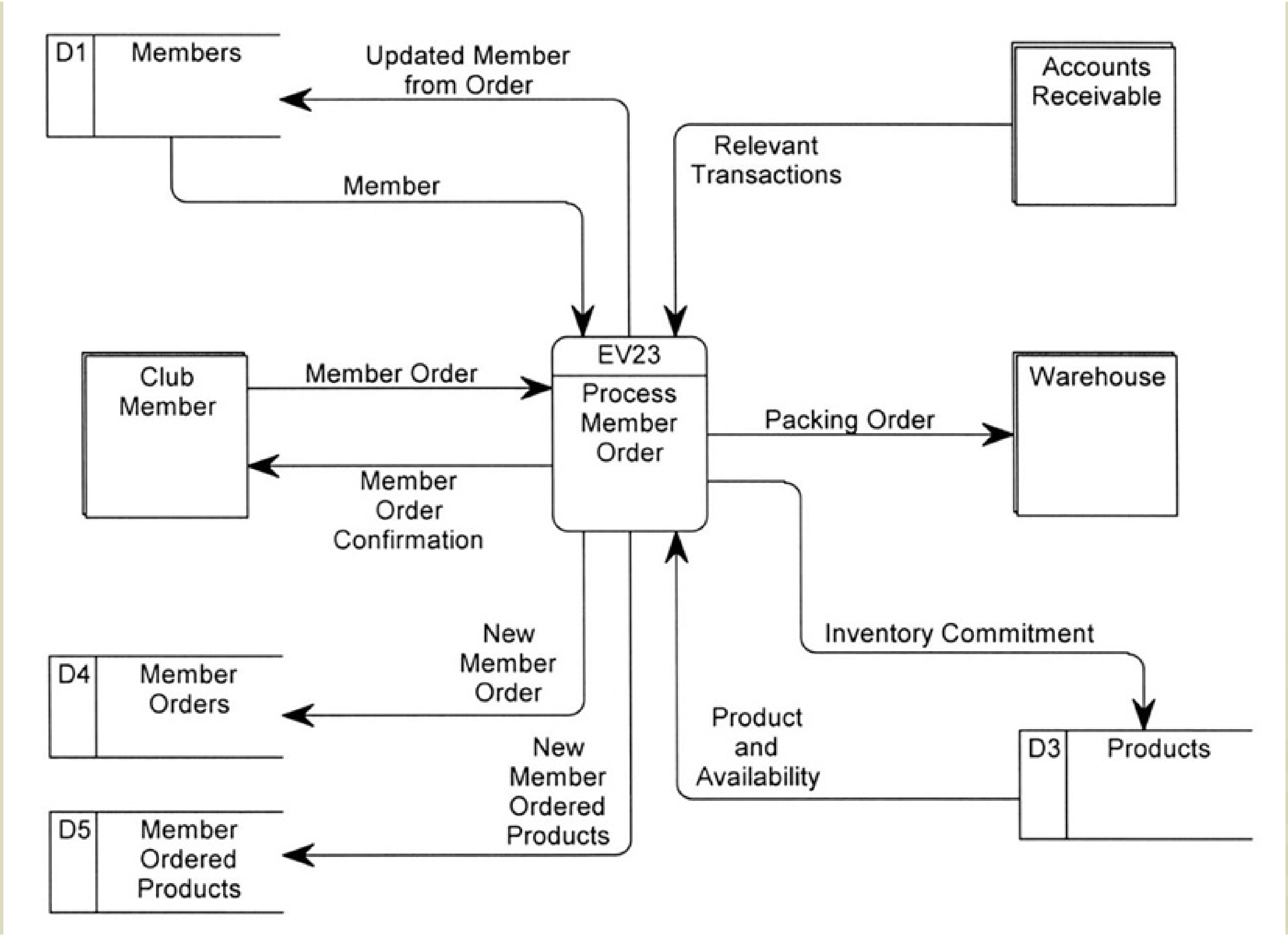
Event diagram – data flow diagram that depicts the context for a single event.

- One diagram for each event process
- Depicts
  - Inputs from external agents
  - Outputs to external agents
  - Data stores from which records must be "read." Data flows should be added and named to reflect the data that is read.
  - Data stores in which records must be created, deleted, or updated. Data flows should be named to reflect the update.

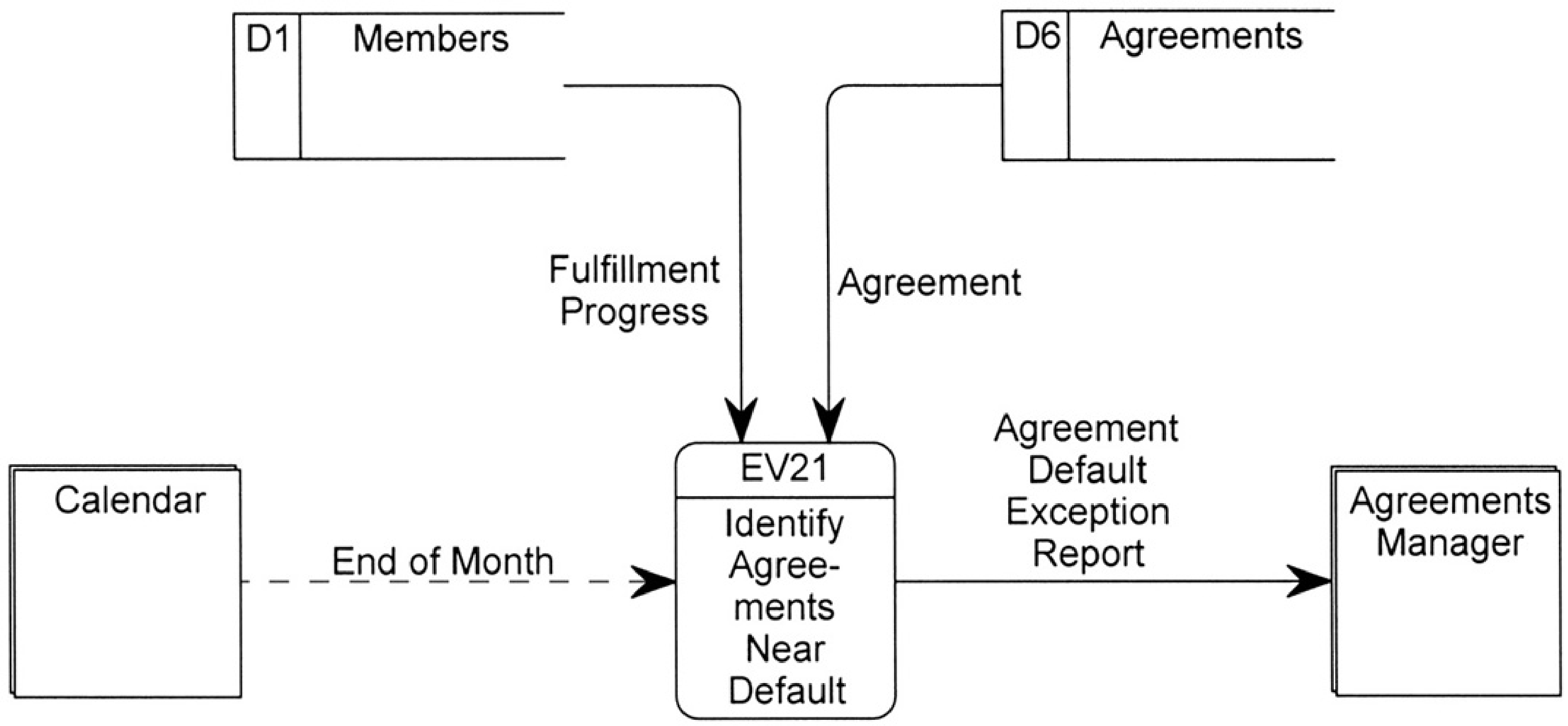
Simple Event Diagram



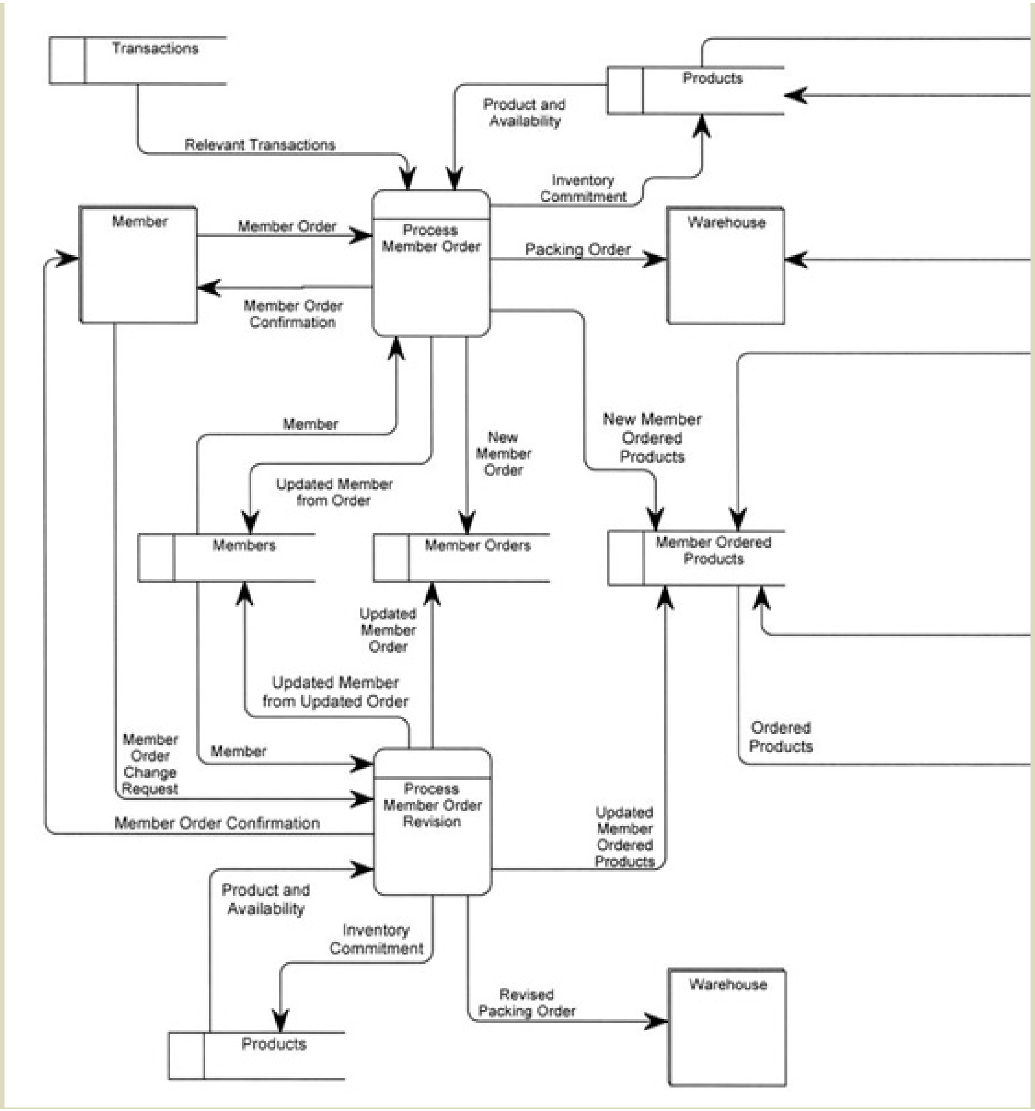
Event Diagram (more complex)



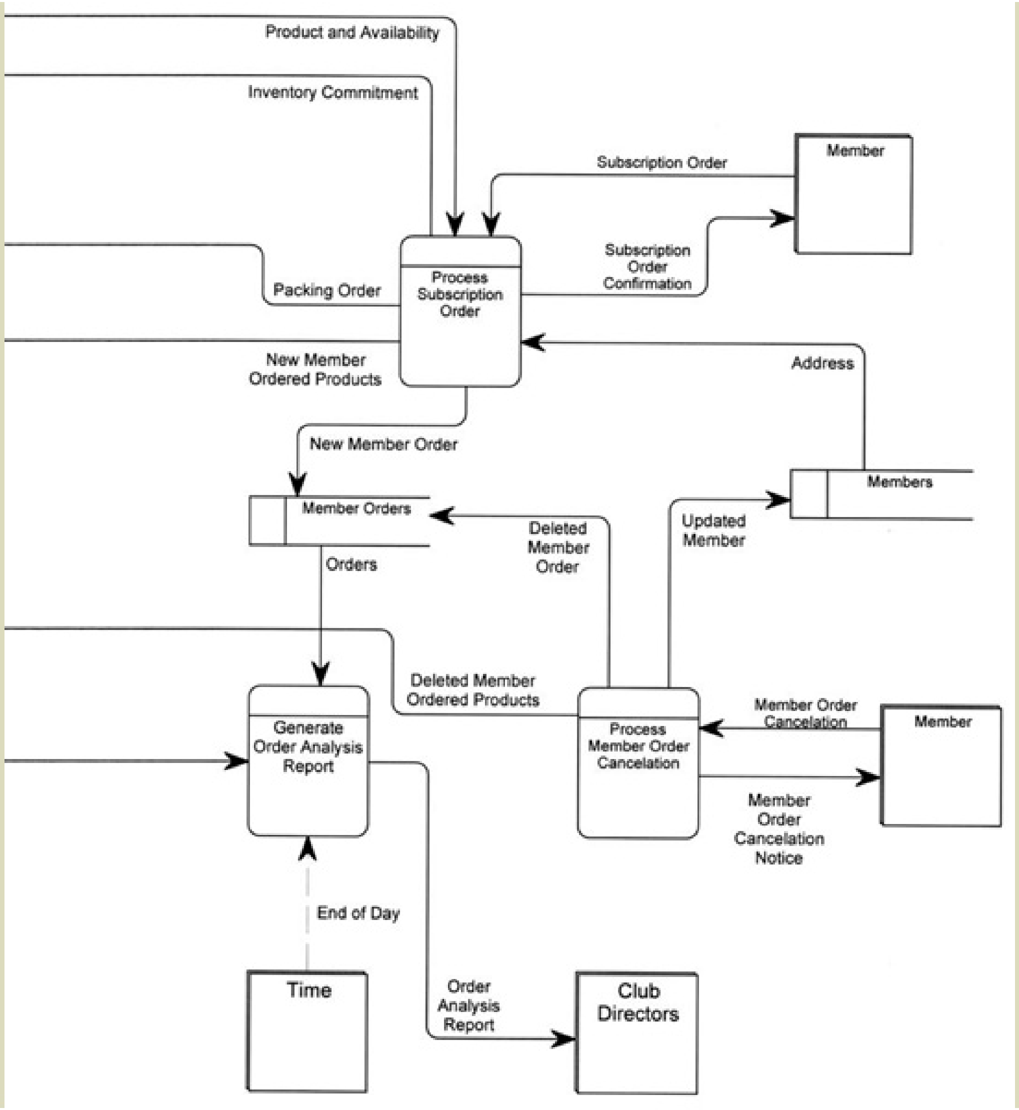
Temporal Event Diagram



System DFD


Transactions
Products
Product and Availability
Relevant Transactions
Inventory Commitment
Member
Member Order
Process Member Order
Warehouse
Packing Order
Member Order Confirmation
Member
New Member Order
New Member Ordered Products
Updated Member from Order
Members
Member Orders
Member Ordered Products
Updated Member Order
Updated Member from Updated Order
Ordered Products
Member Order Change Request
Member
Process Member Order Revision
Member Order Confirmation
Updated Member Ordered Products
Product and Availability
Inventory Commitment
Revised Packing Order
Warehouse
Products

System DFD (concluded)



Balancing

Balancing - a concept that requires that data flow diagrams at different levels of detail reflect consistency and completeness

- Quality assurance technique
- Requires that if you explode a process to another DFD to reveal more detail, you must include the same dta flows and data stores

Primitive Diagrams

- Some (not necessarily all) event processes may be exploded into primitive diagrams to reveal more detail.
  - Complex business transaction processes
  - Process decomposed into multiple elementary processes
  - Each elementary process is cohesive - it does only one thing
  - Flow similar to computer program structure

Process Logic

- Data Flow Diagrams good for identifying and describing processes
- Not good at showing logic inside processes
- Need to specify detailed instructions for elementary processes
- How to do it?
  - Flowcharts & Pseudocode - most end users do not understand them
  - Natural English - imprecise and subject to interpretation

Problems with Natural English

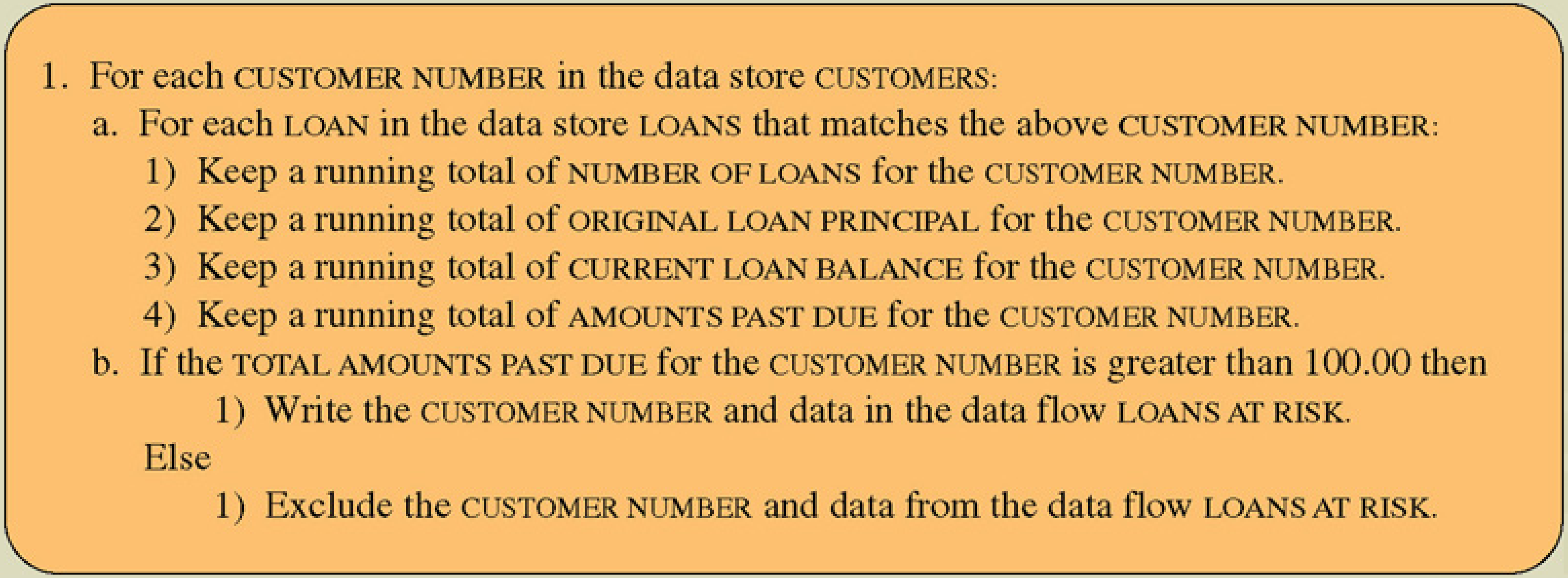
- Many do not write well and do not question writing abilities.
- Many too educated to communicate with general audience
- Some write everything like it was a program.
- Can allow computing jargon, acronyms to dominate language.
- Statements frequently have excessive or confusing scope.
- Overuse compound sentences.
- Too many words have multiple definitions.
- Too many statements use imprecise adjectives.
- Conditional instructions can be imprecise.
- Compound conditions tend to show up in natural English.

Source: Adapted from Matthies, Leslie, *The New Playscript Procedure,* (Stamford, CT: Office Publications, Inc. 1977)

Structured English

Structured English – a language syntax for specifying the logic of a process.

- Based on the relative strengths of structured programming and natural English.

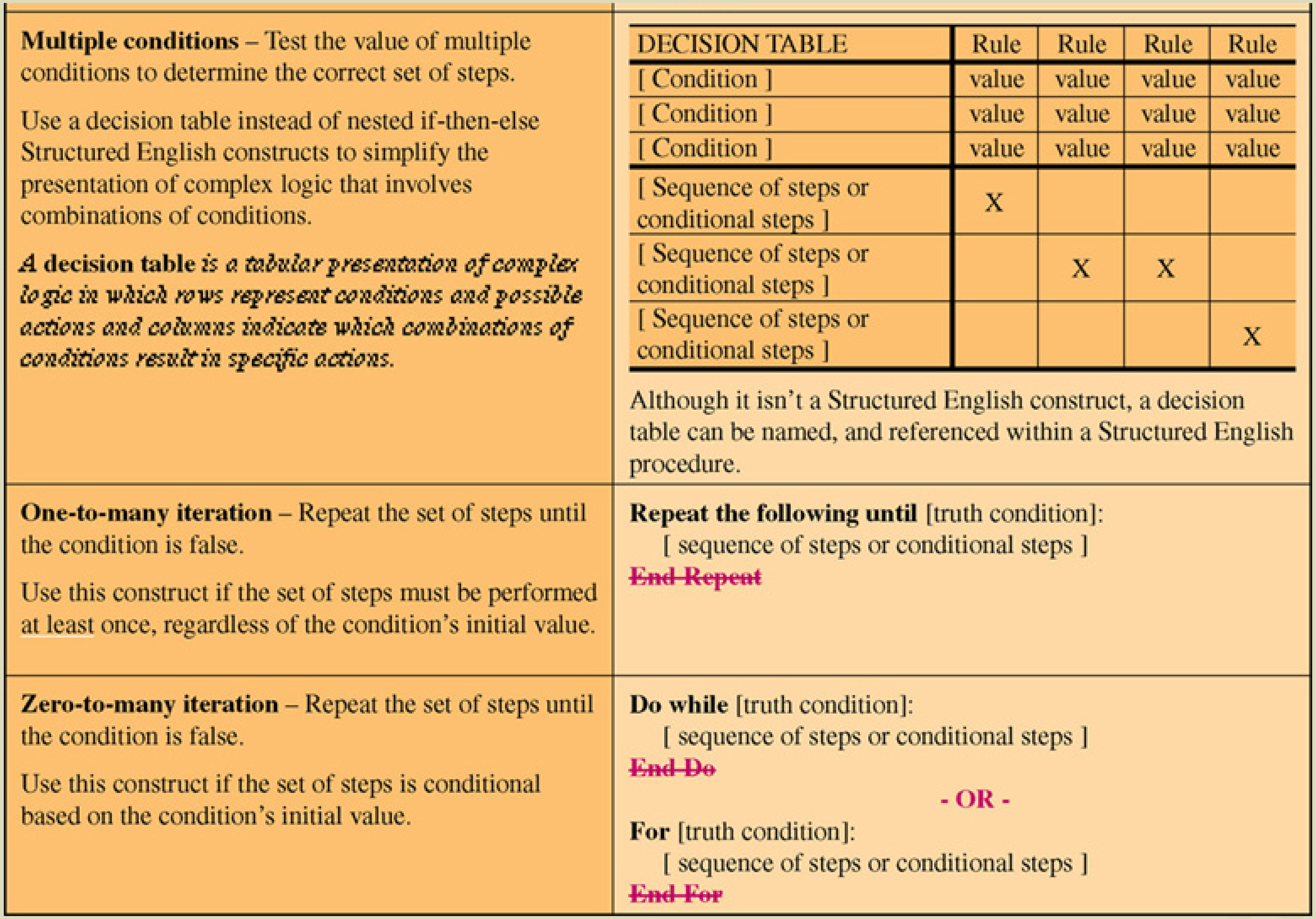


```
1. For each CUSTOMER NUMBER in the data store CUSTOMERS:
   a. For each LOAN in the data store LOANS that matches the above CUSTOMER NUMBER:
      1) Keep a running total of NUMBER OF LOANS for the CUSTOMER NUMBER.
      2) Keep a running total of ORIGINAL LOAN PRINCIPAL for the CUSTOMER NUMBER.
      3) Keep a running total of CURRENT LOAN BALANCE for the CUSTOMER NUMBER.
      4) Keep a running total of AMOUNTS PAST DUE for the CUSTOMER NUMBER.
   b. If the TOTAL AMOUNTS PAST DUE for the CUSTOMER NUMBER is greater than 100.00 then
         1) Write the CUSTOMER NUMBER and data in the data flow LOANS AT RISK.
      Else
         1) Exclude the CUSTOMER NUMBER and data from the data flow LOANS AT RISK.
```

Structured English Constructs (Part 1)

Construct	Sample Template
Sequence of steps – Unconditionally perform a sequence of steps.	[Step 1] [Step 2] … [Step n]
Simple condition steps – If the specified condition is true, then perform the first set of steps. Otherwise, perform the second set of steps. Use this construct if the condition has only two possible values. (Note: The second set of conditions is optional.)	**If** [truth condition] **then** [sequence of steps or other conditional steps] **else** [sequence of steps or other conditional steps] ~~**End If**~~
Complex condition steps – Test the value of the condition and perform the appropriate set of steps. Use this construct if the condition has more than two values.	**Do the following based on** [condition]: **Case 1: If** [condition] = [value] then [sequence of steps or other conditional steps] **Case 2: If** [condition] = [value] then [sequence of steps or other conditional steps] … **Case n: If** [condition] = [value] then [sequence of steps or other conditional steps] ~~**End Case**~~

Structured English Constructs (Part 2)



Multiple conditions – Test the value of multiple conditions to determine the correct set of steps.

Use a decision table instead of nested if-then-else Structured English constructs to simplify the presentation of complex logic that involves combinations of conditions.

*A **decision table** is a tabular presentation of complex logic in which rows represent conditions and possible actions and columns indicate which combinations of conditions result in specific actions.*

DECISION TABLE	Rule	Rule	Rule	Rule
[Condition]	value	value	value	value
[Condition]	value	value	value	value
[Condition]	value	value	value	value
[Sequence of steps or conditional steps]	X			
[Sequence of steps or conditional steps]		X	X	
[Sequence of steps or conditional steps]				X

Although it isn't a Structured English construct, a decision table can be named, and referenced within a Structured English procedure.

One-to-many iteration – Repeat the set of steps until the condition is false.

Use this construct if the set of steps must be performed at least once, regardless of the condition's initial value.

```
Repeat the following until [truth condition]:
    [ sequence of steps or conditional steps ]
~~End Repeat~~
```

Zero-to-many iteration – Repeat the set of steps until the condition is false.

Use this construct if the set of steps is conditional based on the condition's initial value.

```
Do while [truth condition]:
    [ sequence of steps or conditional steps ]
~~End Do~~
                    - OR -
For [truth condition]:
    [ sequence of steps or conditional steps ]
~~End For~~
```

Structured English Restrictions on Process Logic

- Only strong, imperative verbs may be used.
- Only names that have been defined in project dictionary may be used.
- Formulas should be stated clearly using appropriate mathematical notations.
- Undefined adjectives and adverbs are not permitted.
- Blocking and indentation are used to set off the beginning and ending of constructs.
- User readability should always take priority.

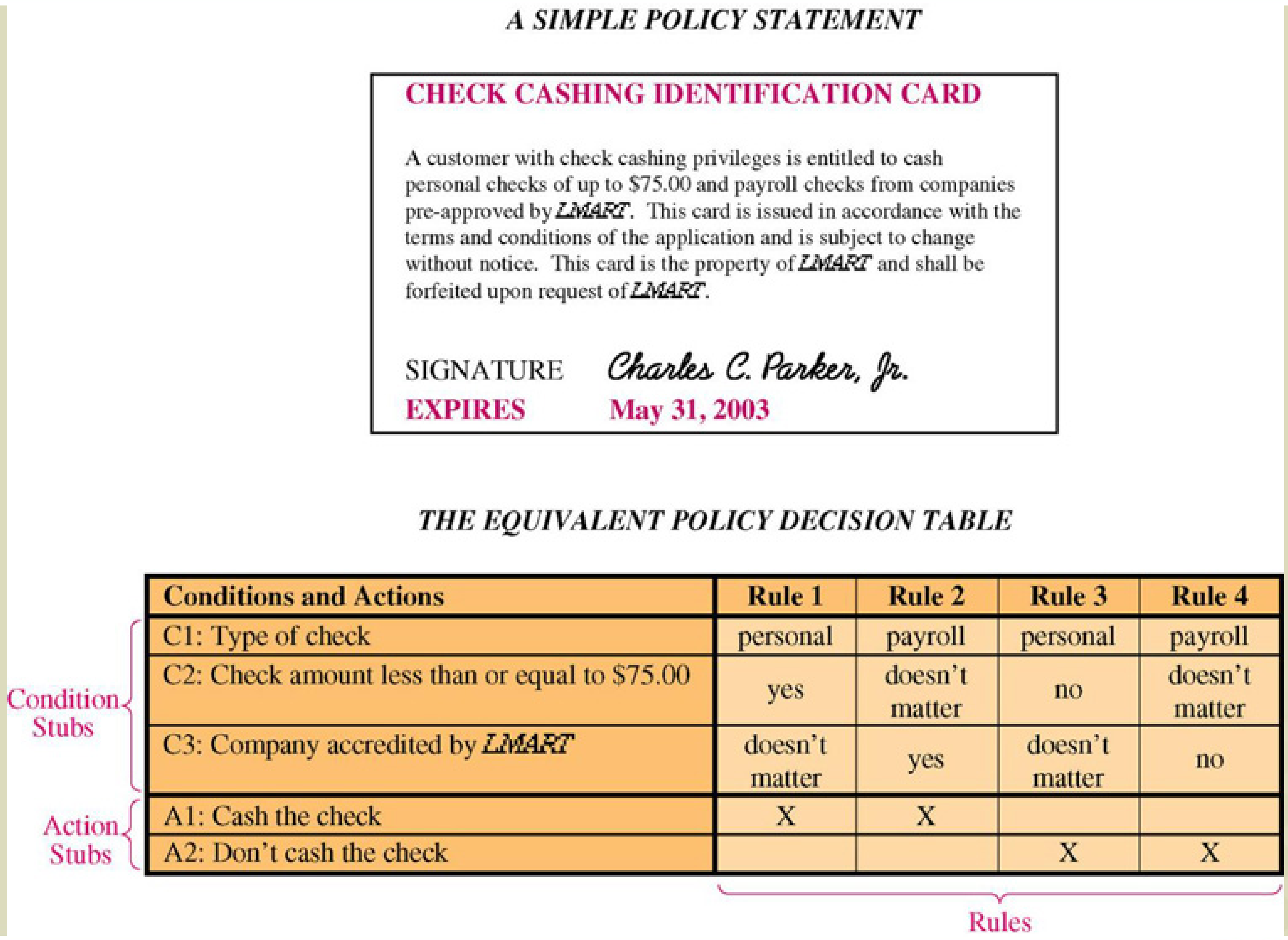
Policies and Decision Tables

Policy – a set of rules that govern show a process is to be completed.

Decision table – a tabular form of presentation that specifies a set of conditions and their corresponding actions.

- As required to implement a policy.

A Simple Decision Table



A SIMPLE POLICY STATEMENT

CHECK CASHING IDENTIFICATION CARD

A customer with check cashing privileges is entitled to cash personal checks of up to $75.00 and payroll checks from companies pre-approved by *LMART*. This card is issued in accordance with the terms and conditions of the application and is subject to change without notice. This card is the property of *LMART* and shall be forfeited upon request of *LMART*.

SIGNATURE Charles C. Parker, Jr.

EXPIRES **May 31, 2003**

THE EQUIVALENT POLICY DECISION TABLE

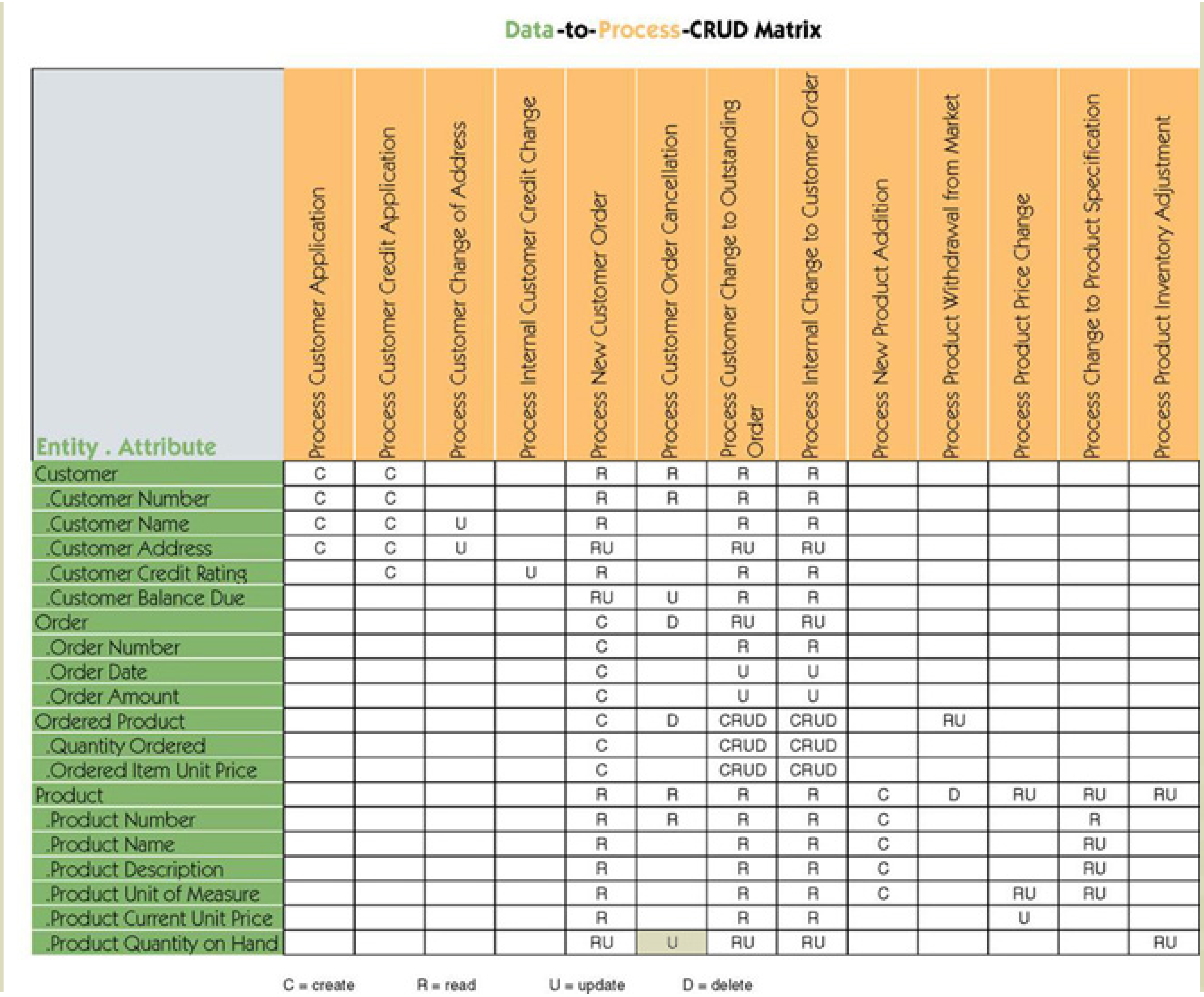
Conditions and Actions	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**
C1: Type of check	personal	payroll	personal	payroll
C2: Check amount less than or equal to $75.00	yes	doesn't matter	no	doesn't matter
C3: Company accredited by *LMART*	doesn't matter	yes	doesn't matter	no
A1: Cash the check	X	X		
A2: Don't cash the check			X	X

Condition Stubs

Action Stubs

Rules

Data & Process Model Synchronization CRUD Matrix

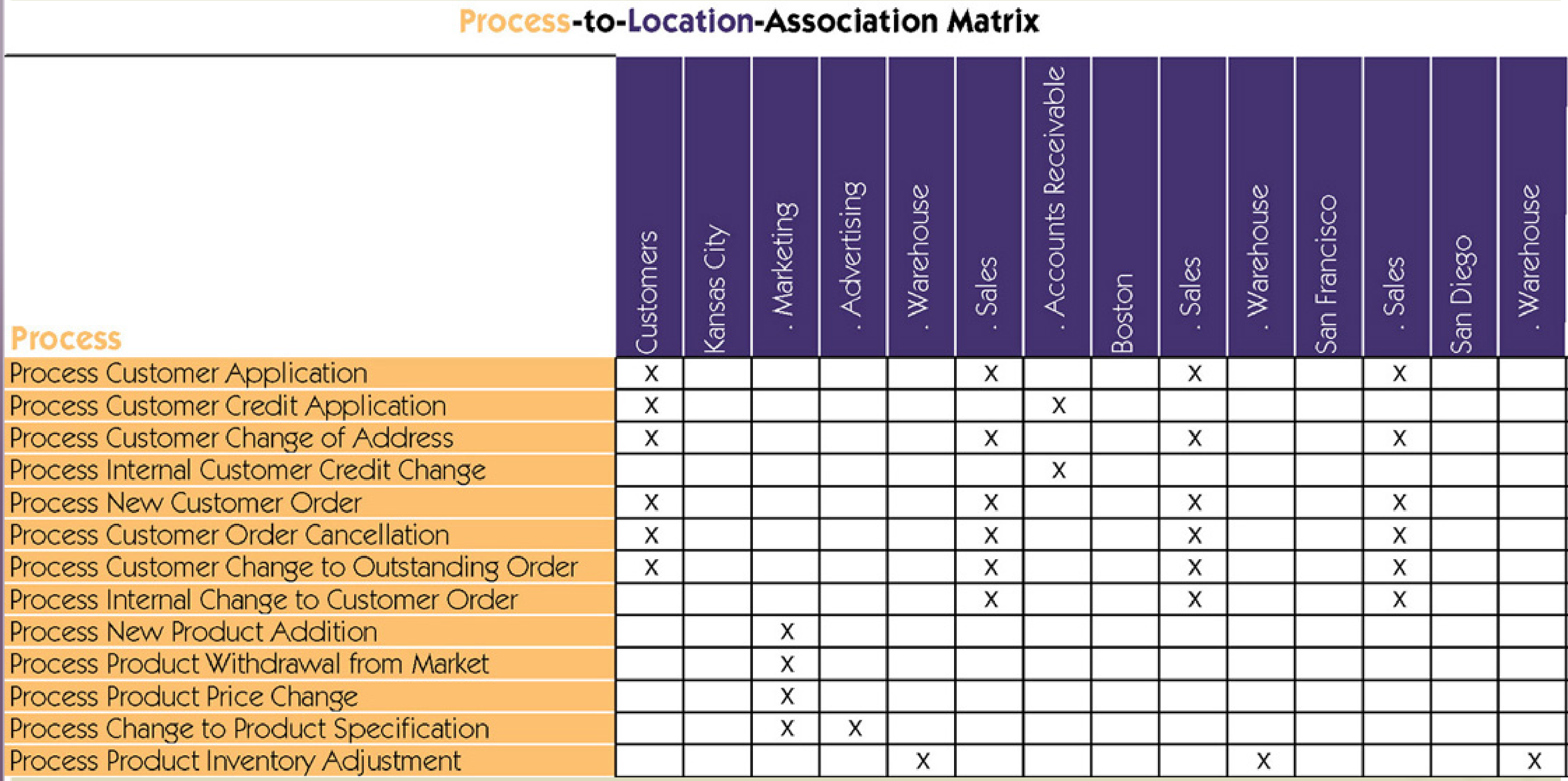


Data-to-Process-CRUD Matrix

Entity . Attribute	Process Customer Application	Process Customer Credit Application	Process Customer Change of Address	Process Internal Customer Credit Change	Process New Customer Order	Process Customer Order Cancellation	Process Customer Change to Outstanding Order	Process Internal Change to Customer Order	Process New Product Addition	Process Product Withdrawal from Market	Process Product Price Change	Process Change to Product Specification	Process Product Inventory Adjustment
Customer	C	C			R	R	R	R					
.Customer Number	C	C			R	R	R	R					
.Customer Name	C	C	U		R		R	R					
.Customer Address	C	C	U		RU		RU	RU					
.Customer Credit Rating		C		U	R		R	R					
.Customer Balance Due					RU	U	R	R					
Order					C	D	RU	RU					
.Order Number					C		R	R					
.Order Date					C		U	U					
.Order Amount					C		U	U					
Ordered Product					C	D	CRUD	CRUD		RU			
.Quantity Ordered					C		CRUD	CRUD					
.Ordered Item Unit Price					C		CRUD	CRUD					
Product					R	R	R	R	C	D	RU	RU	RU
.Product Number					R	R	R	R	C			R	
.Product Name					R		R	R	C			RU	
.Product Description					R		R	R	C			RU	
.Product Unit of Measure					R		R	R	C		RU	RU	
.Product Current Unit Price					R		R	R			U		
.Product Quantity on Hand					RU	U	RU	RU					RU

C = create R = read U = update D = delete

Process Distribution



Process-to-Location-Association Matrix

Process	Customers	Kansas City	. Marketing	. Advertising	. Warehouse	. Sales	. Accounts Receivable	Boston	. Sales	. Warehouse	San Francisco	. Sales	San Diego	. Warehouse
Process Customer Application	X					X			X			X		
Process Customer Credit Application	X						X							
Process Customer Change of Address	X					X			X			X		
Process Internal Customer Credit Change							X							
Process New Customer Order	X					X			X			X		
Process Customer Order Cancellation	X					X			X			X		
Process Customer Change to Outstanding Order	X					X			X			X		
Process Internal Change to Customer Order						X			X			X		
Process New Product Addition			X											
Process Product Withdrawal from Market			X											
Process Product Price Change			X											
Process Change to Product Specification			X	X										
Process Product Inventory Adjustment					X					X				X