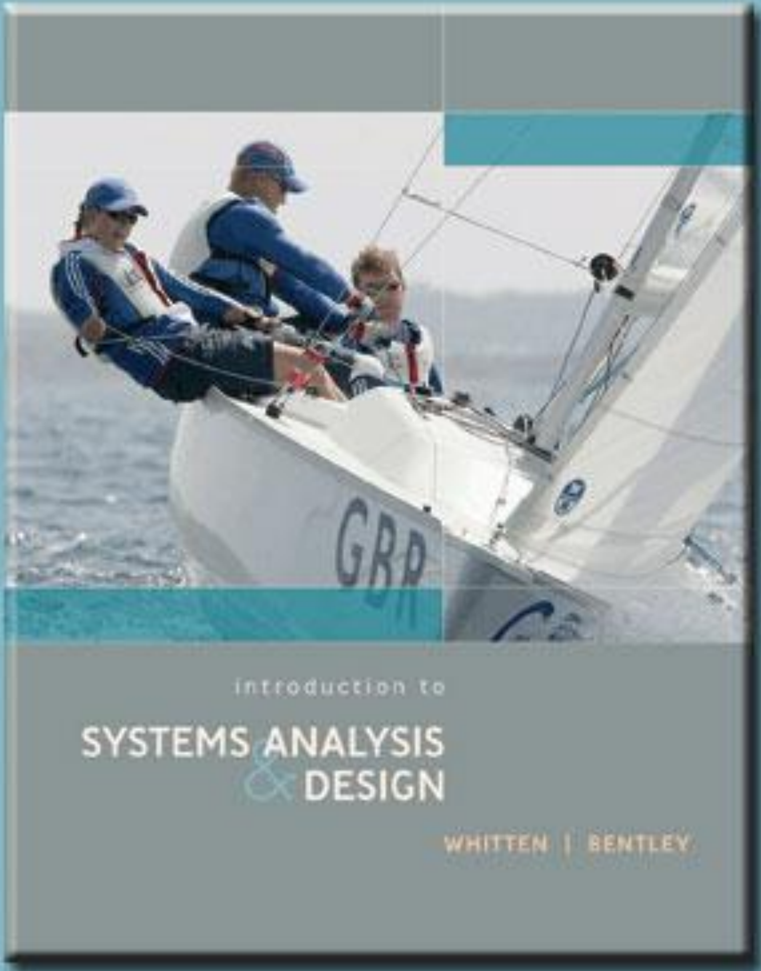


Chapter 8

Process Modeling

Objectives

- 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 (Structured English and decision tables), respectively.

Process Modeling and DFDs

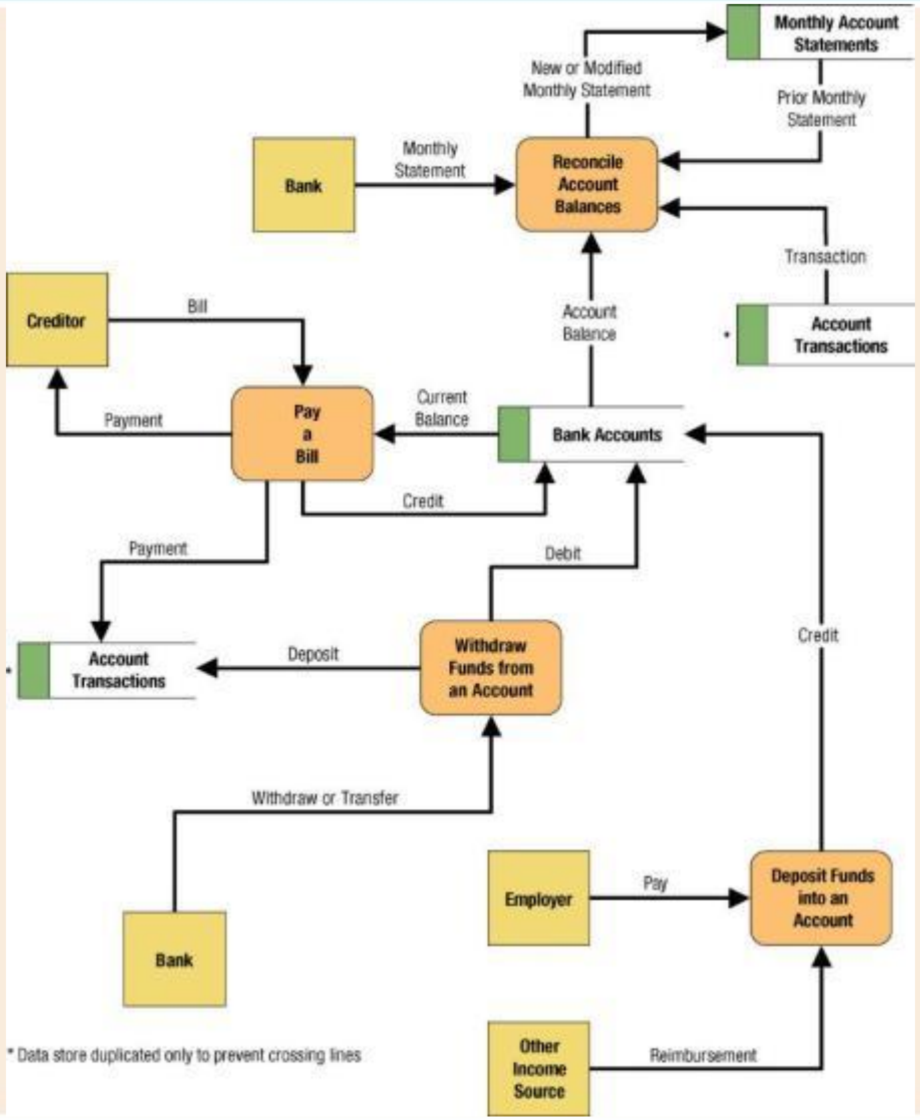
Process modeling – a technique used to organize and document a system's processes.

- 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



External Agents

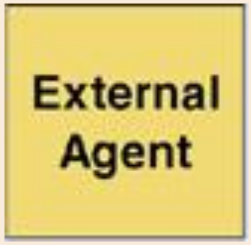
External agent – an outside person, unit, system, or organization that interacts with a system. Also called an *external entity*.

– External agents define the "boundary" or scope of a system being modeled.

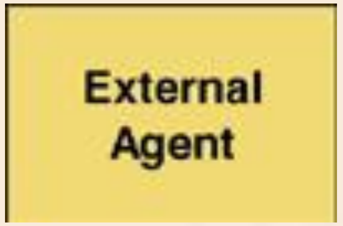
– Almost always one of the following:

- 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

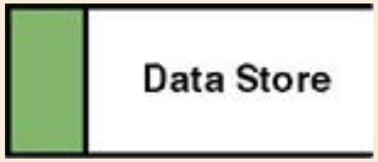


DeMarco/Yourdon shape

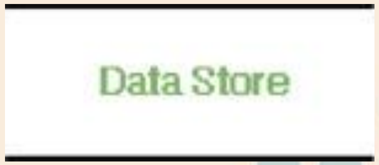
Data Stores

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

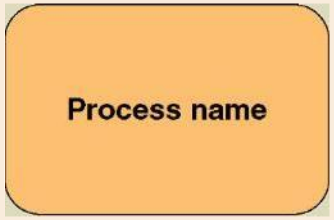


DeMarco/Yourdon shape

Process Concepts

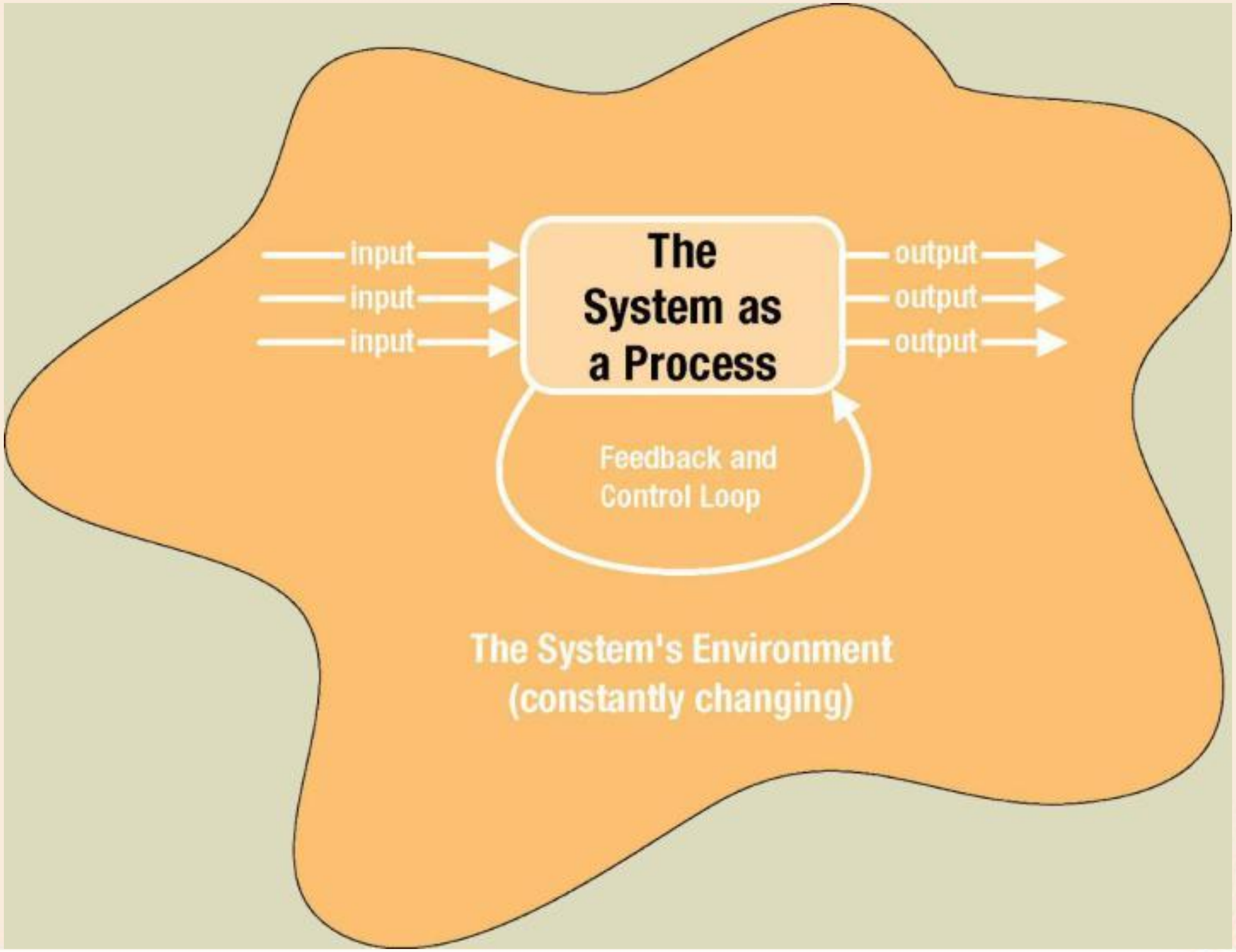
Process – work performed by a system in response to incoming data flows or conditions. A synonym is *transform*.

- 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 .



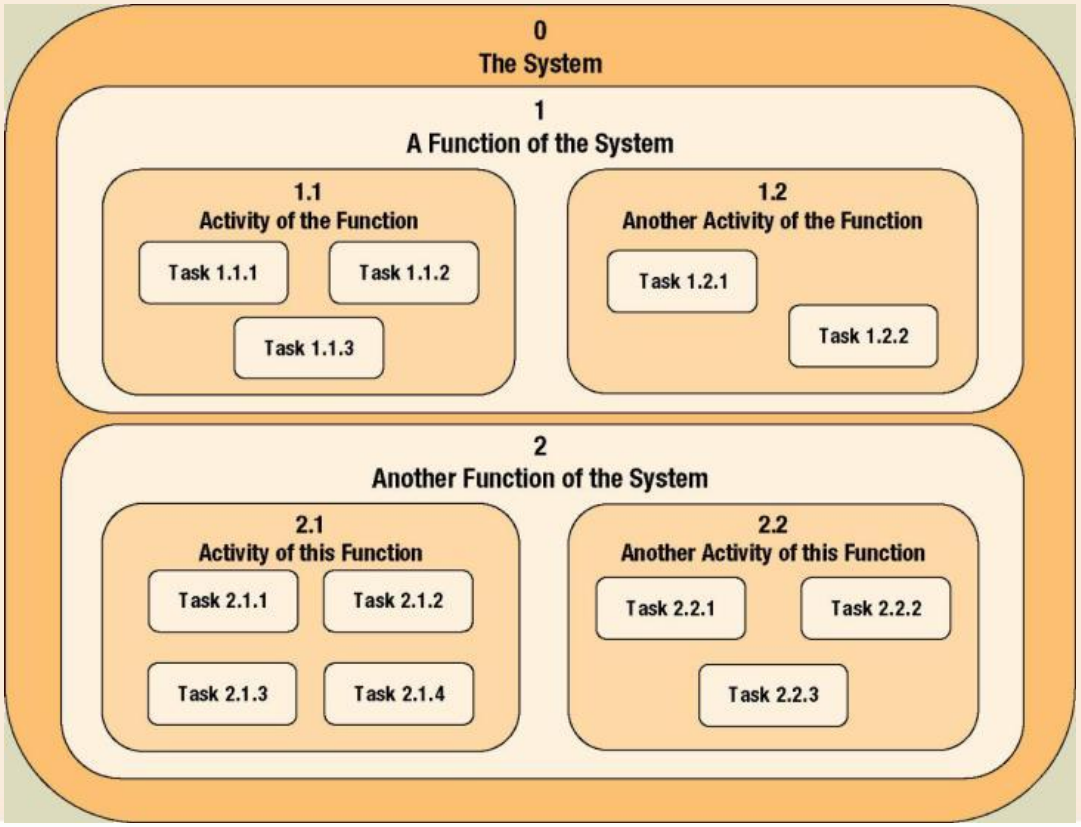
Gane and Sarson shape

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)

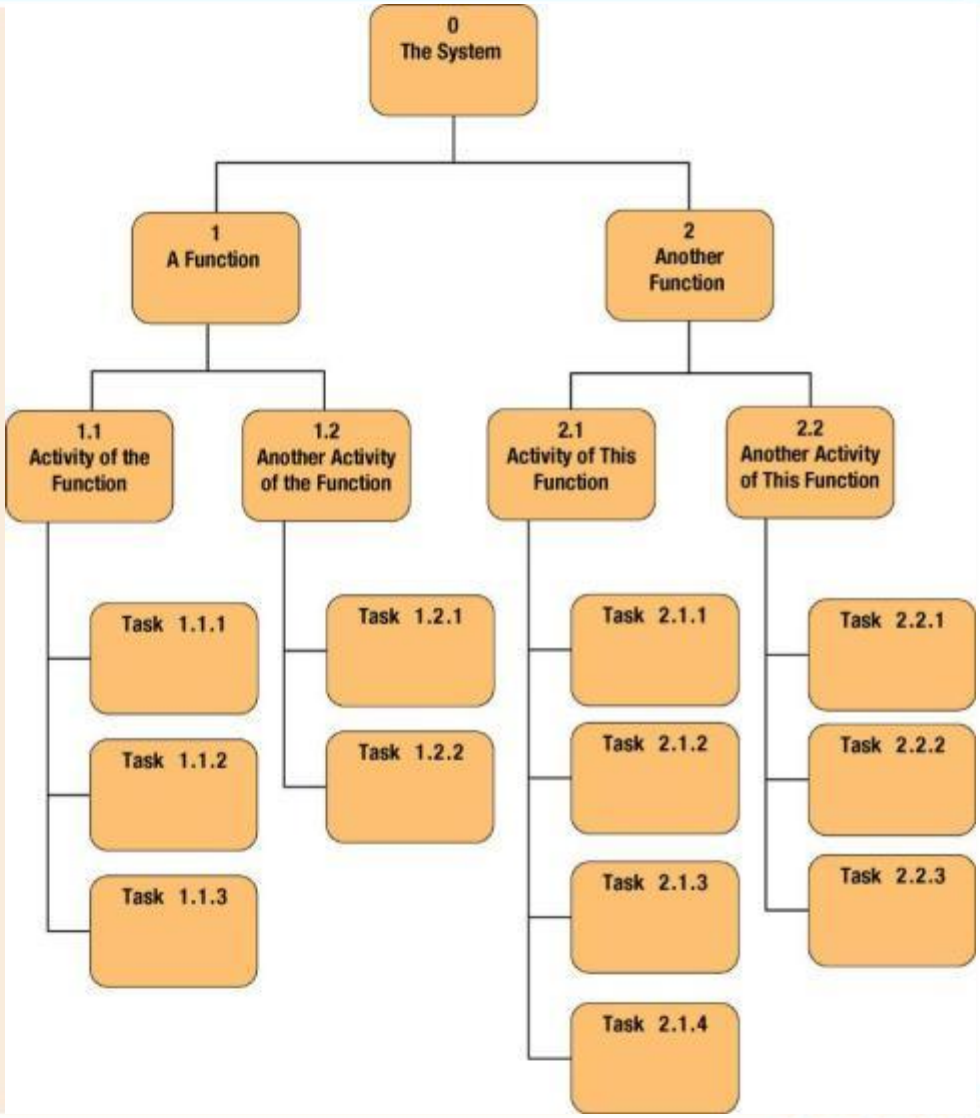
Process Decomposition

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.

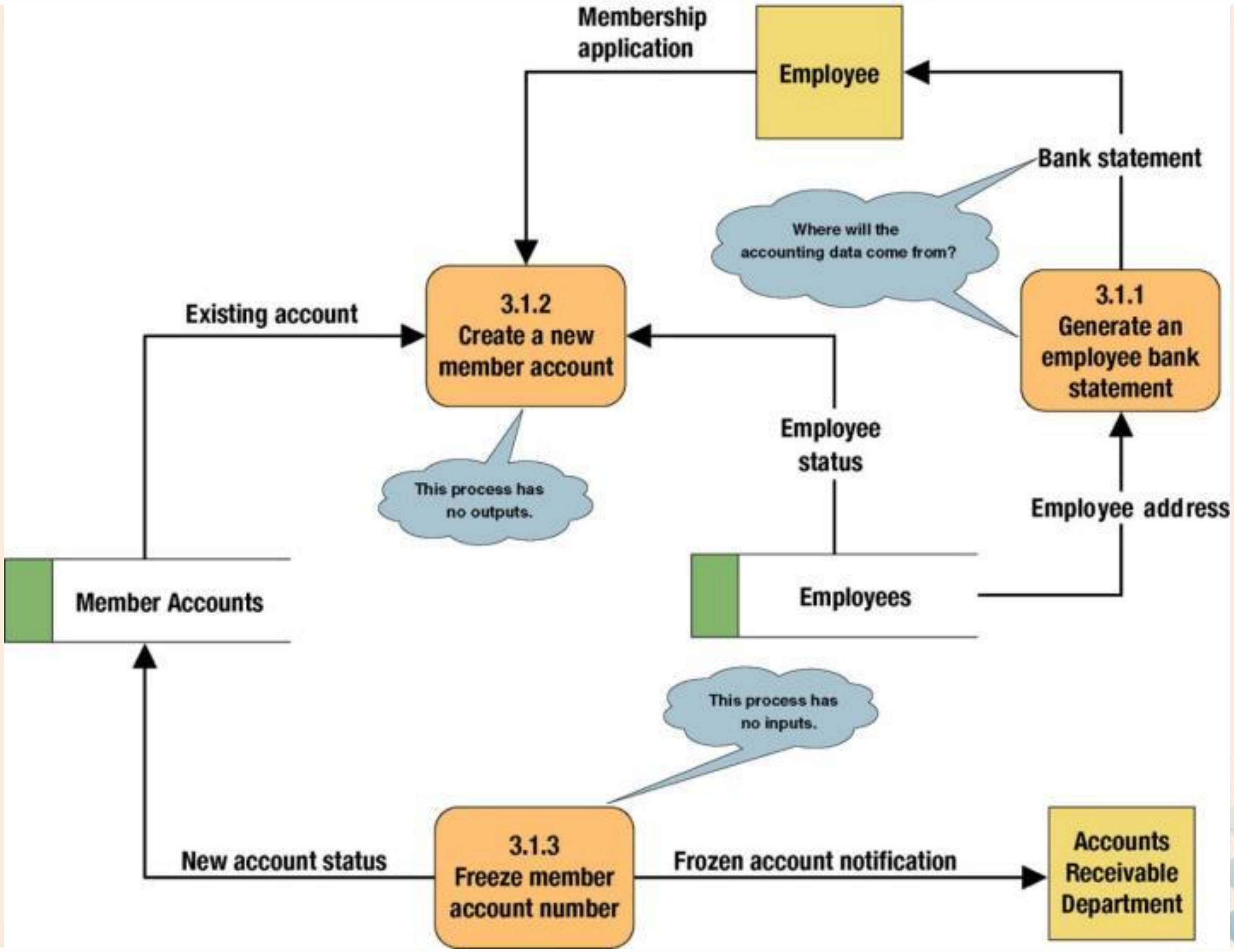
Event – a logical unit of work that must be completed as a whole. Sometimes called a *transaction*.

- 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.

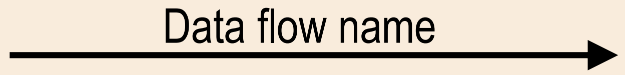
Common Process Errors on DFDs


Membership application
Employee
Bank statement
Where will the accounting data come from?
3.1.1 Generate an employee bank statement
Existing account
3.1.2 Create a new member account
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.

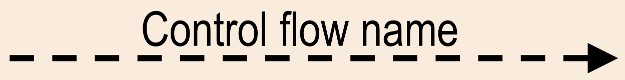
- A data flow is data in motion
- A data flow may also be used to represent the creation, reading, deletion, or updating of data in a file or database (called a data store).



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

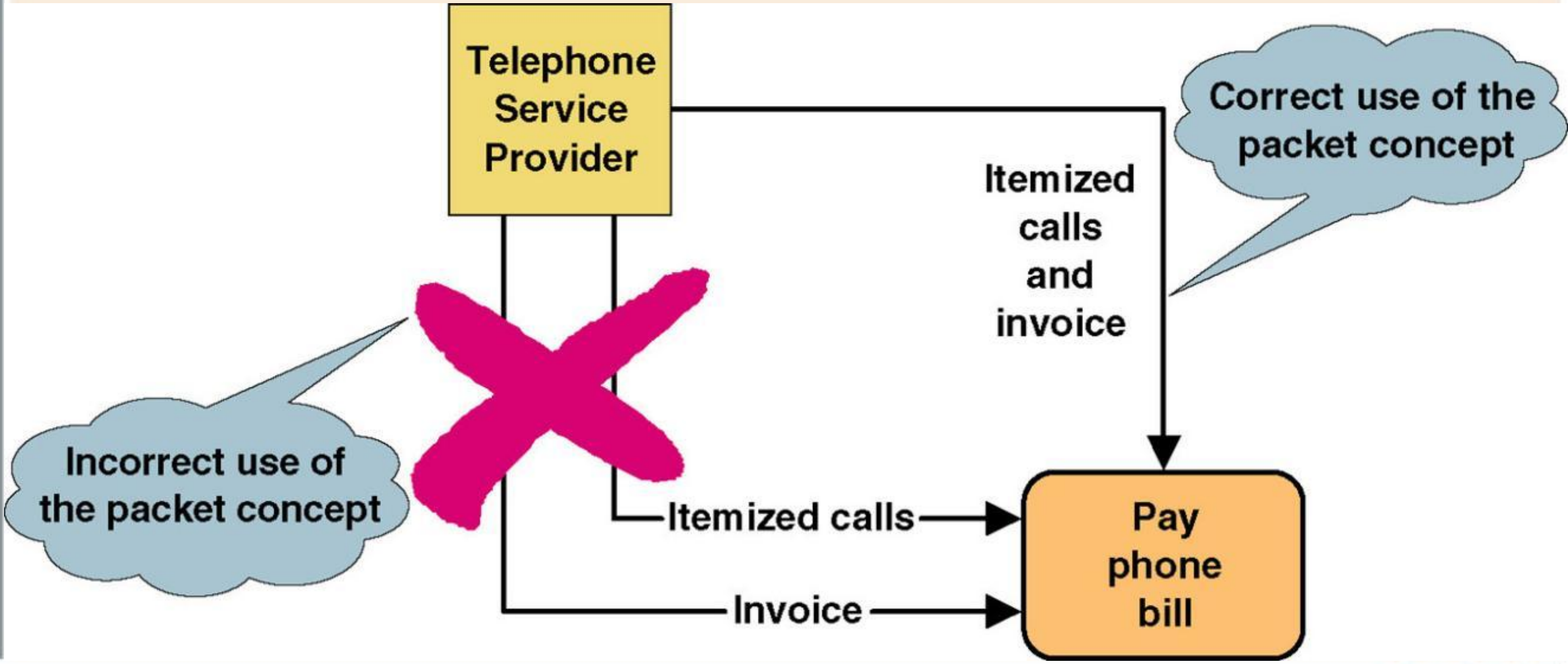
Control flow – a condition or nondata event that triggers a process.

- Used sparingly on DFDs.

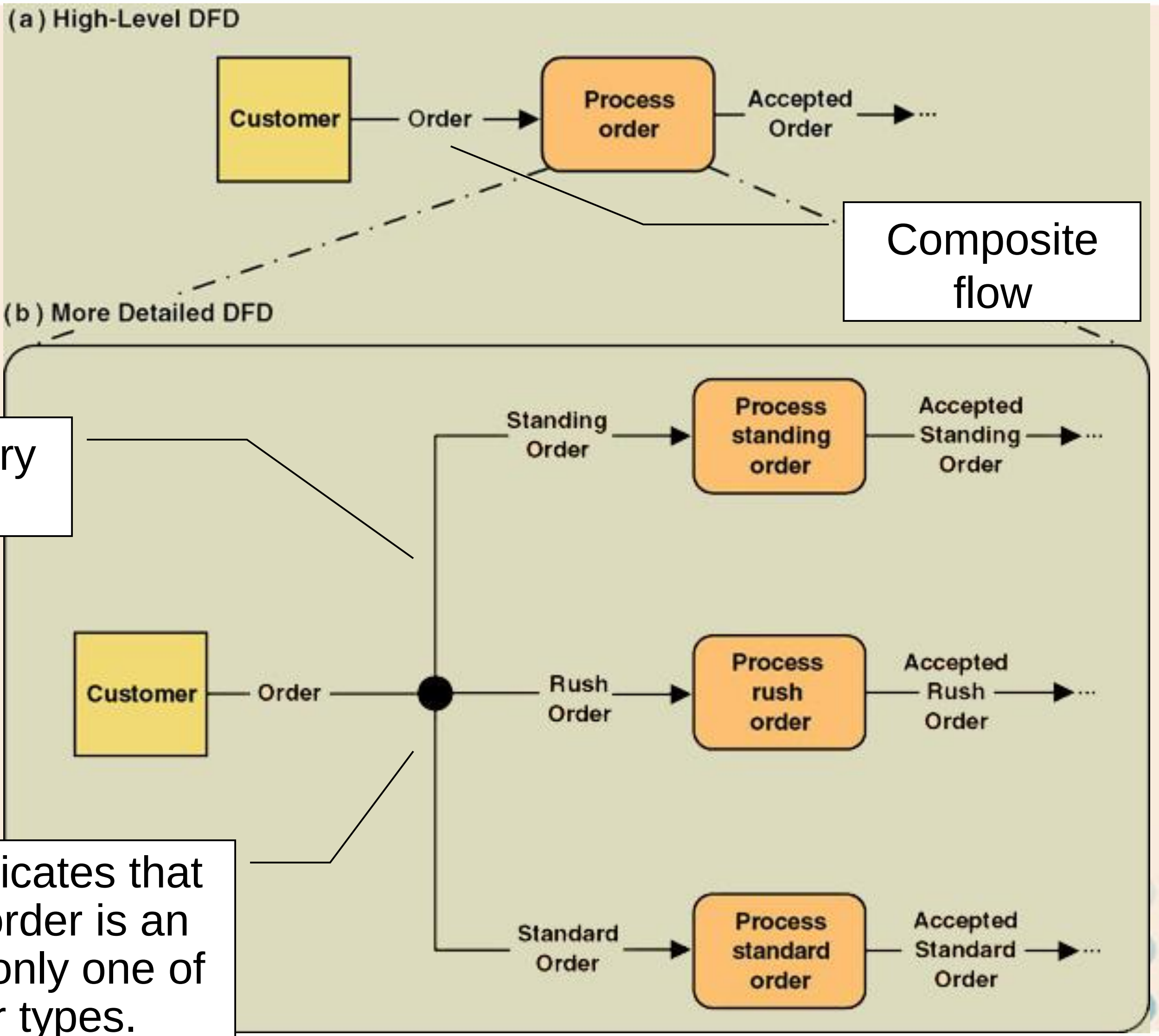


Data Flow Packet Concept

- 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

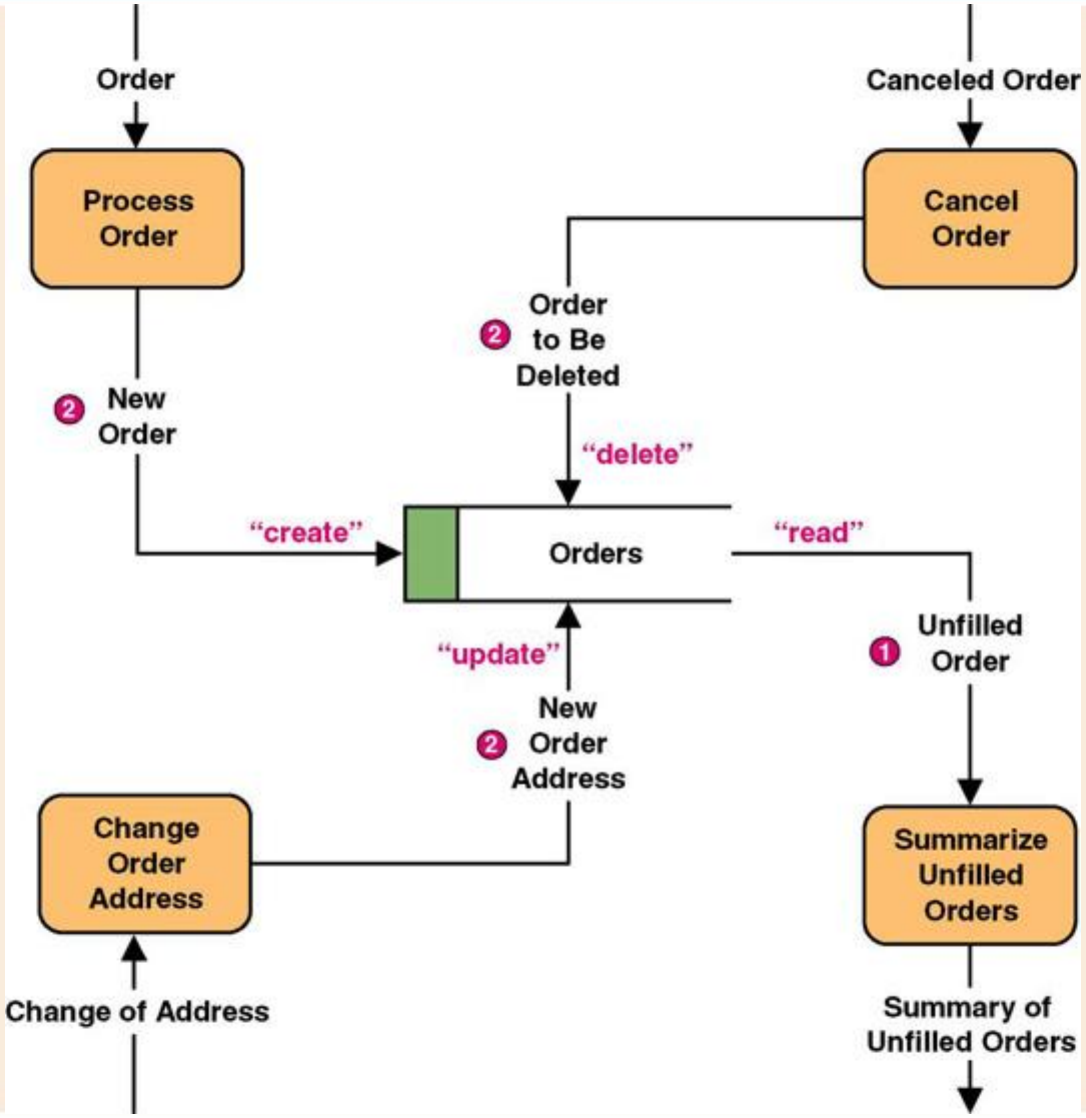


Composite flow

Elementary flows

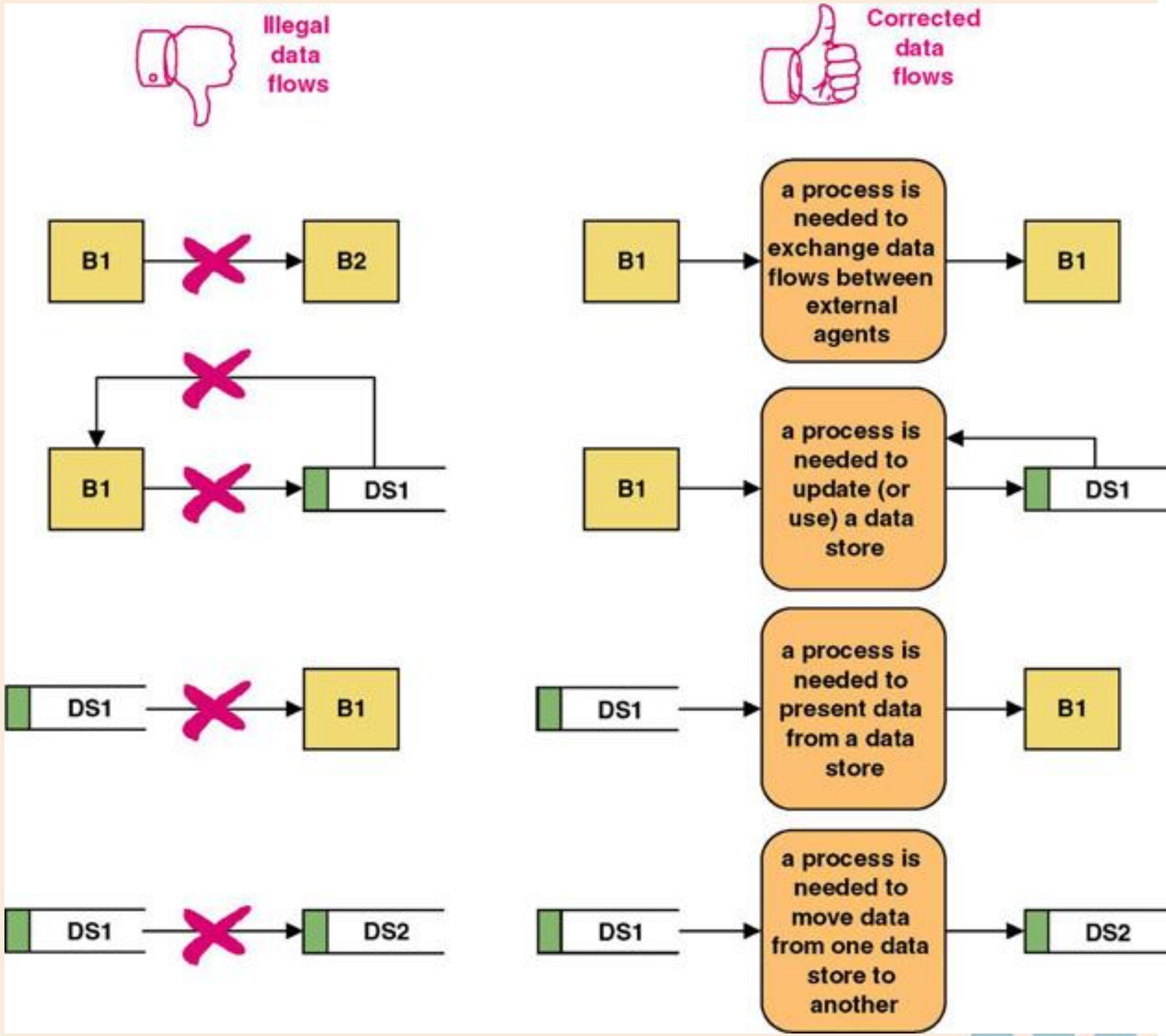
Junction indicates that any given order is an instance of only one of the order types.

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

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 Structures

Data attribute – the smallest piece of data that has meaning to the users and the business.

Data structure – a specific arrangement of data attributes that defines an instance of a data flow.

- The data attributes that comprise a data flow are organized into data structures.
- Data flows can be described in terms of the following types of data structures:
  - A *sequence* or group of data attributes that occur one after another.
  - The *selection* of one or more attributes from a set of attributes.
  - The *repetition* of one or more attributes.

Data Structure for a Data Flow

DATA STRUCTURE	ENGLISH ENTERPRETATION
ORDER= ORDER NUMBER + ORDER DATE+ [PERSONAL CUSTOMER NUMBER, CORPORATE ACCOUNT NUMBER]+ SHIPPING ADDRESS=ADDRESS+ (BILLING ADDRESS=ADDRESS)+ 1 {PRODUCT NUMBER+ PRODUCT DESCRIPTION+ QUANTITY ORDERED+ PRODUCT PRICE+ PRODUCT PRICE SOURCE+ EXTENDED PRICE } N+ SUM OF EXTENDED PRICES+ PREPAID AMOUNT+ (CREDIT CARD NUMBER+EXPIRATION DATE) (QUOTE NUMBER)	An instance of ORDER consists of: ORDER NUMBER and ORDER DATE and Either PERSONAL CUSTOMER NUMBER or CORPORATE ACCOUNT NUMBER and SHIPPING ADDRESS (which is equivalent to ADDRESS) and optionally: BILLING ADDRESS (which is equivalent to ADDRESS) and one or more instances of: PRODUCT NUMBER and PRODUCT DESCRIPTION and QUANTITY ORDERED and PRODUCT PRICE and PRODUCT PRICE SOURCE and EXTENDED PRICE and SUM OF EXTENDED PRICES and PREPAID AMOUNT and optionally: both CREDIT CARD NUMBER and EXPIRATION DATE
ADDRESS= (POST OFFICE BOX NUMBER)+ STREET ADDRESS+ CITY+ [STATE, MUNICIPALITY]+ (COUNTRY)+ POSTAL CODE	An instance of ADDRESS consists of: optionally: POST OFFICE BOX NUMBER and STREET ADDRESS and CITY and Either STATE or MUNICIPALITY and optionally: COUNTRY and POSTAL CODE

Data Structure Constructs

Data Structure	**Format by Example** (relevant portion is **boldfaced**	**English Interpretation** (relevant portion is **boldfaced**)
Sequence of Attributes - The sequence data structure indicates one or more attributes that may (or must) be included in a data flow.	WAGE AND TAX STATEMENT= **TAXPAYER IDENTIFICATION NUMBER+** **TAXPAYER NAME+** **TAXPAYER ADDRESS+** **WAGES, TIPS, AND COMPENSATION+** **FEDERAL TAX WITHHELD+…**	An instance of WAGE AND TAX STATEMENTS consists of: **TAXPAYER IDENTIFICATION NUMBER and** **TAXPAYER NAME and** **TAXPAYER ADDRESS and** **WAGES, TIPS AND COMPENSATION and** **FEDERAL TAX WITHHELD and…**
Selection of Attributes - The selection data structure allows you to show situations where different sets of attributes describe different instances of the data flow.	ORDER= **(PERSONAL CUSTOMER NUMBER,** **CORPORATE ACCOUNT NUMBER)+** ORDER DATE+…	An instance or ORDER consists of: **Either PERSONAL CUSTOMER NUMBER or** **CORPORATE ACCOUNT NUMBER;** and ORDER DATE and…

Data Structure Constructs (continued)

Data Structure	**Format by Example** (relevant portion is **boldfaced**	**English Interpretation** (relevant portion is **boldfaced**)
Repetition of Attributes - The repetition data structure is used to set off a data attribute or group of data attributes that may (or must) repeat themselves a specific number of time for a single instance of the data flow. The minimum number of repetitions is usually *zero* or *one*. The maximum number of repetitions may be specified as "n" meaning "many" where the actual number of instances varies for each instance of the data flow.	POLICY NUMBER+ POLICYHOLDER NAME+ POLICY HOLDER ADDRESS+ **0 {DEPENDENT NAME+** **DEPENDENT'S RELATIONSHIP} N+** **1 {EXPENSE DESCRIPTION+** **SERVICE PROVIDER+** **EXPENSE AMOUNT} N**	An instance of CLAIM consists of: POLICY NUMBER and POLICYHOLDER NAME and POLICYHOLDER ADDRESS and **zero or more instance of:** **DEPENDENT NAME and** **DEPENDENT'S RELATIONSHIP and** **one or more instances of:** **EXPENSE DESCRIPTION and** **SERVICE PROVIDER and** **EXPENSE ACCOUNT**

Data Structure Constructs (concluded)

Data Structure	**Format by Example** (relevant portion is **boldfaced**	**English Interpretation** (relevant portion is **boldfaced**)
Optional Attributes - The optional notation indicates that an attribute, or group of attributes <u>in a sequence or selection date structure</u> may not be included in all instances of a data flow. *Note: For the repetition data structure, a minimum of "zero" is the same as making the entire repeating group "optional."*	CLAIM= POLICY NUMBER+ POLICYHOLDER NAME+ POLICYHOLDER ADDRESS+ **(SPOUSE NAME+** **DATE OF BIRTH)**+…	An instance of CLAIM consists of: POLICY NUMBER and POLICYHOLDER NAME and POLICYHOLDER ADDRESS and **optionally, SPOUSE NAME and** **DATE OF BIRTH** and…

Reusable Attributes - For groups of attributes that are contained in many data flows, it is desirable to create a separate data structure that can be reused in other data structures.	DATE=
	MONTH+
	DAY+
	YEAR+
		Then, the reusable structures can be included in other data flow structures as follows:
		ORDER=ORDER NUMBER…+DATE
		INVOICE=INVOICE NUMBER…+DATE
		PAYMENT=CUSTOMER NUMBER…+DATE

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

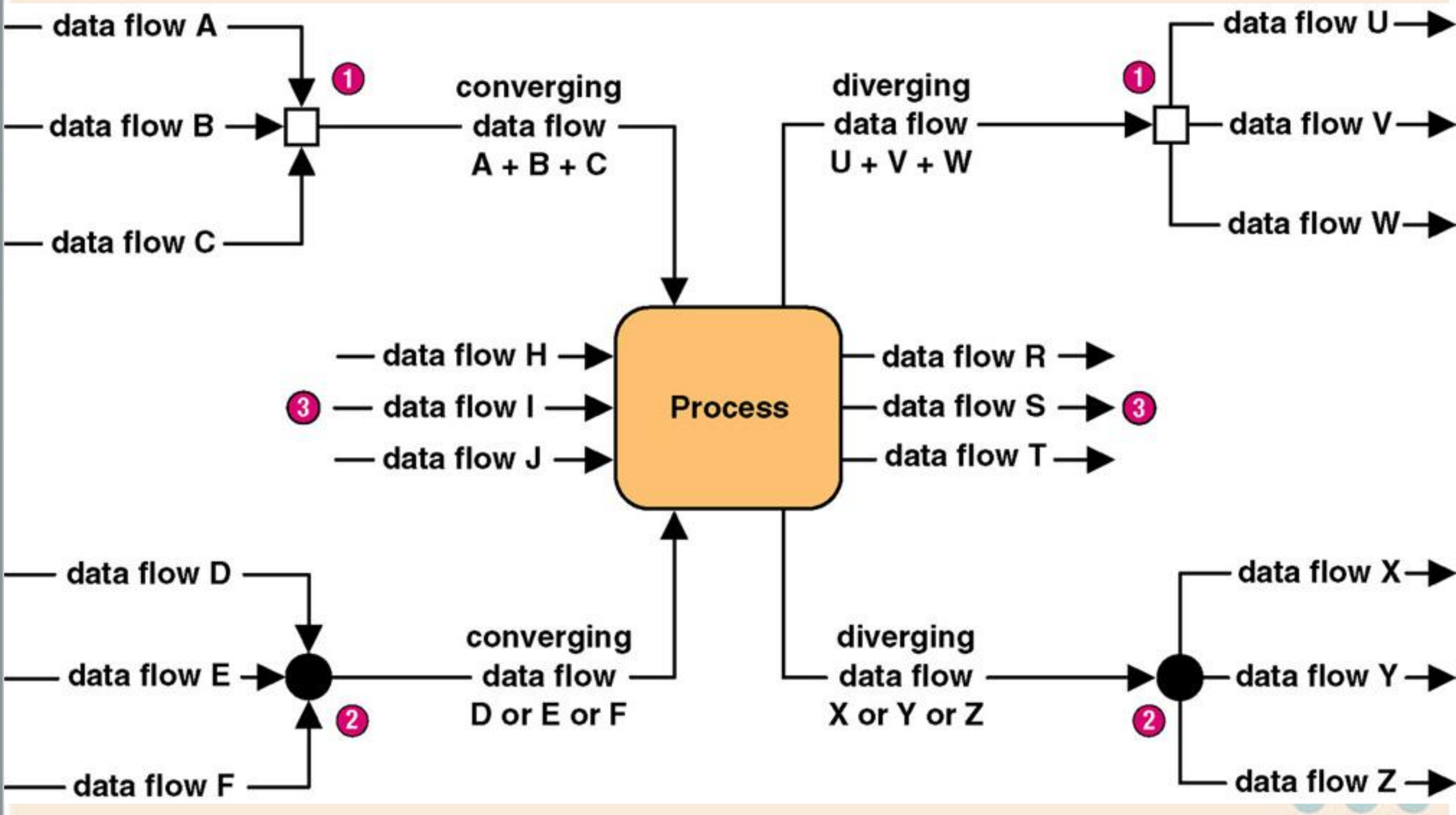
Diverging data flow – a data flow that splits into multiple data flows.

- 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

- 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

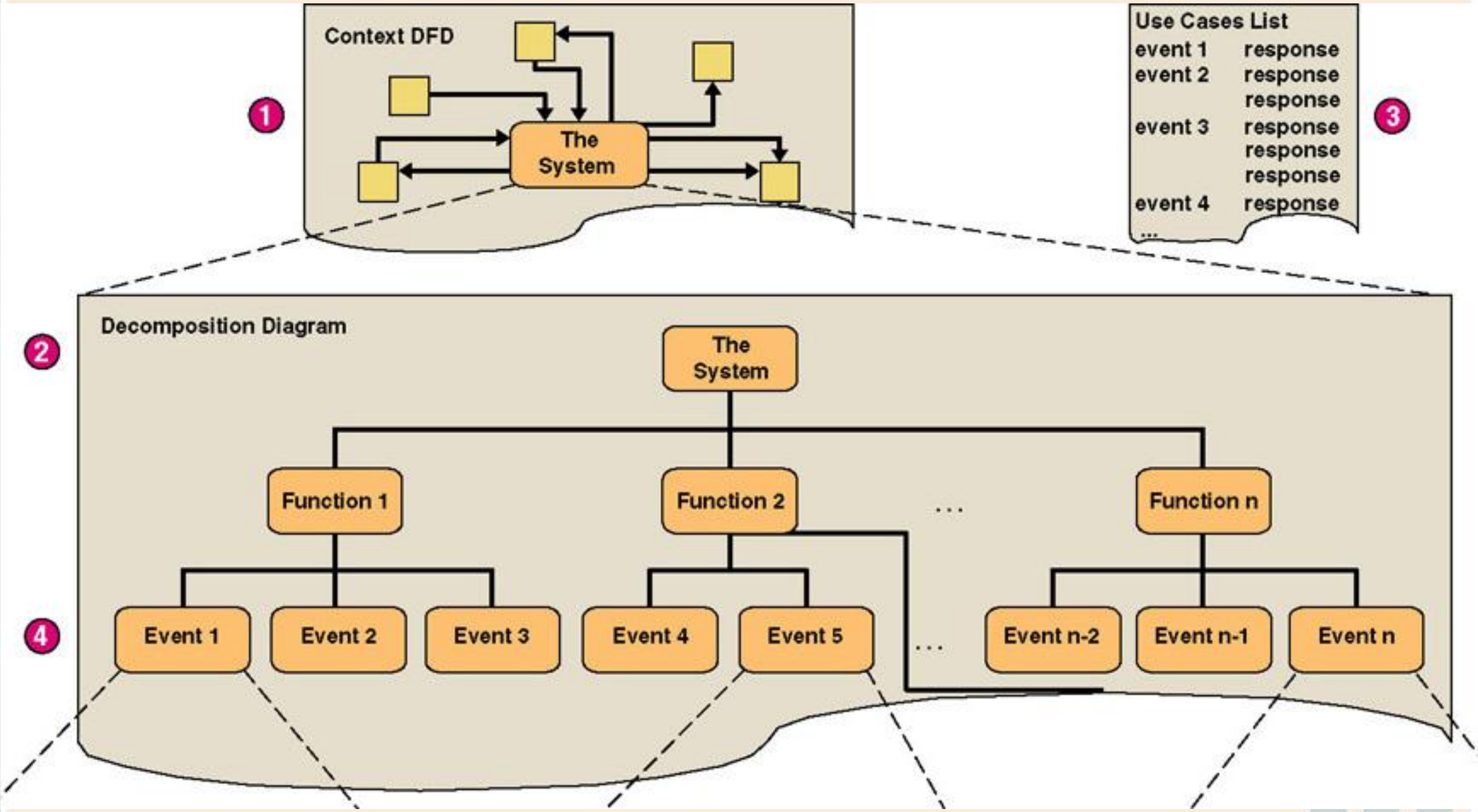
Rarely practiced anymore because cumbersome & time-consuming

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.

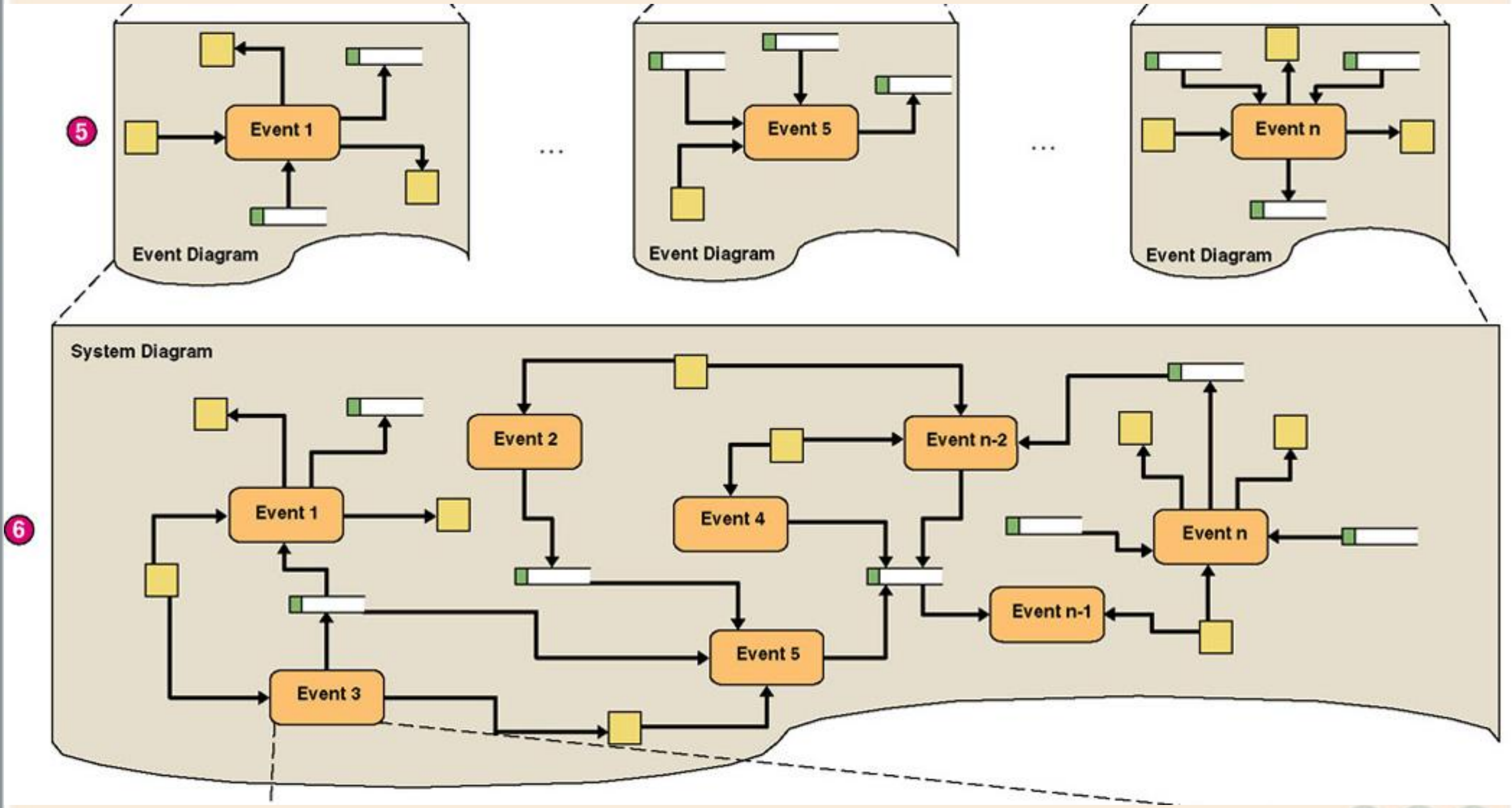
Modern Structured Analysis (More Commonly Practiced)

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.

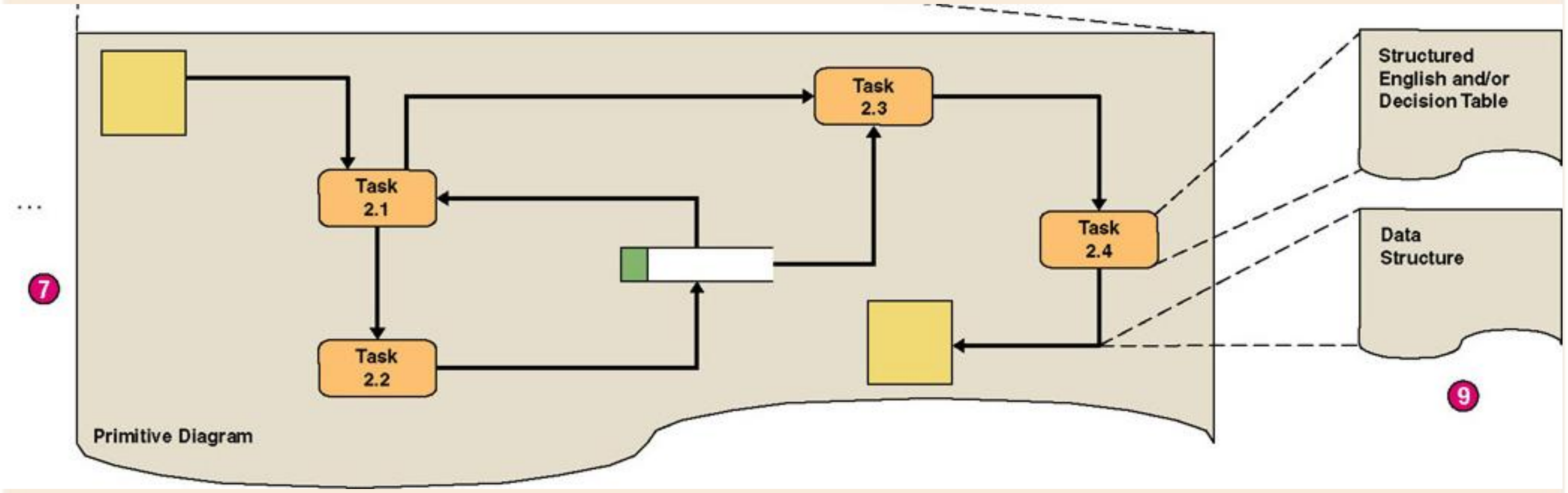
Structured Analysis Diagram Progression (1 of 3)



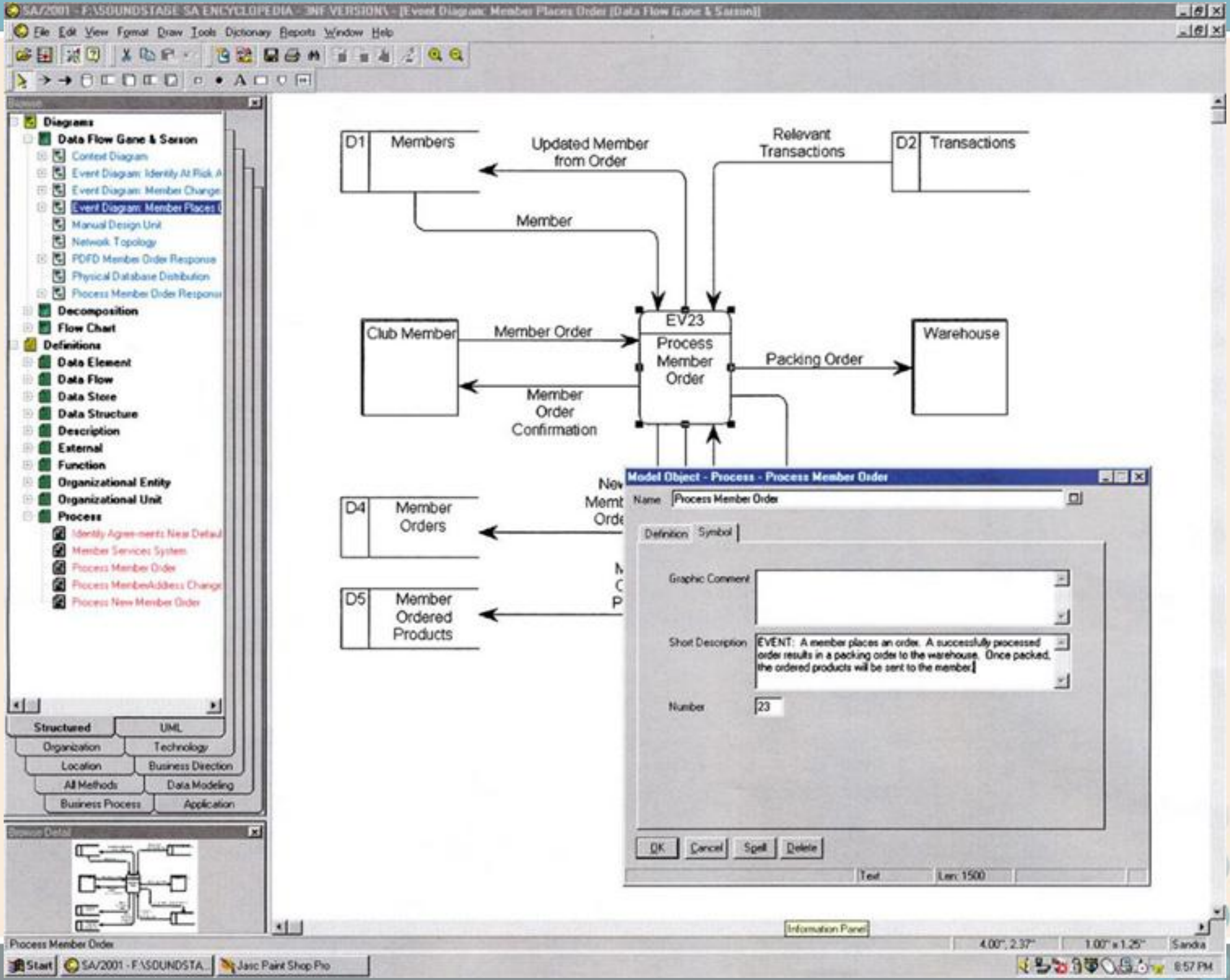
Structured Analysis Diagram Progression (2 of 3)



Structured Analysis Diagram Progression (3 of 3)



CASE for Process Modeling



Model Object - Process - Process Member Order

Name: Process Member Order

Definition | Symbol

Graphic Comment:

Short Description: EVENT: A member places an order. A successfully processed order results in a packing order to the warehouse. Once packed, the ordered products will be sent to the member.

Number: 23

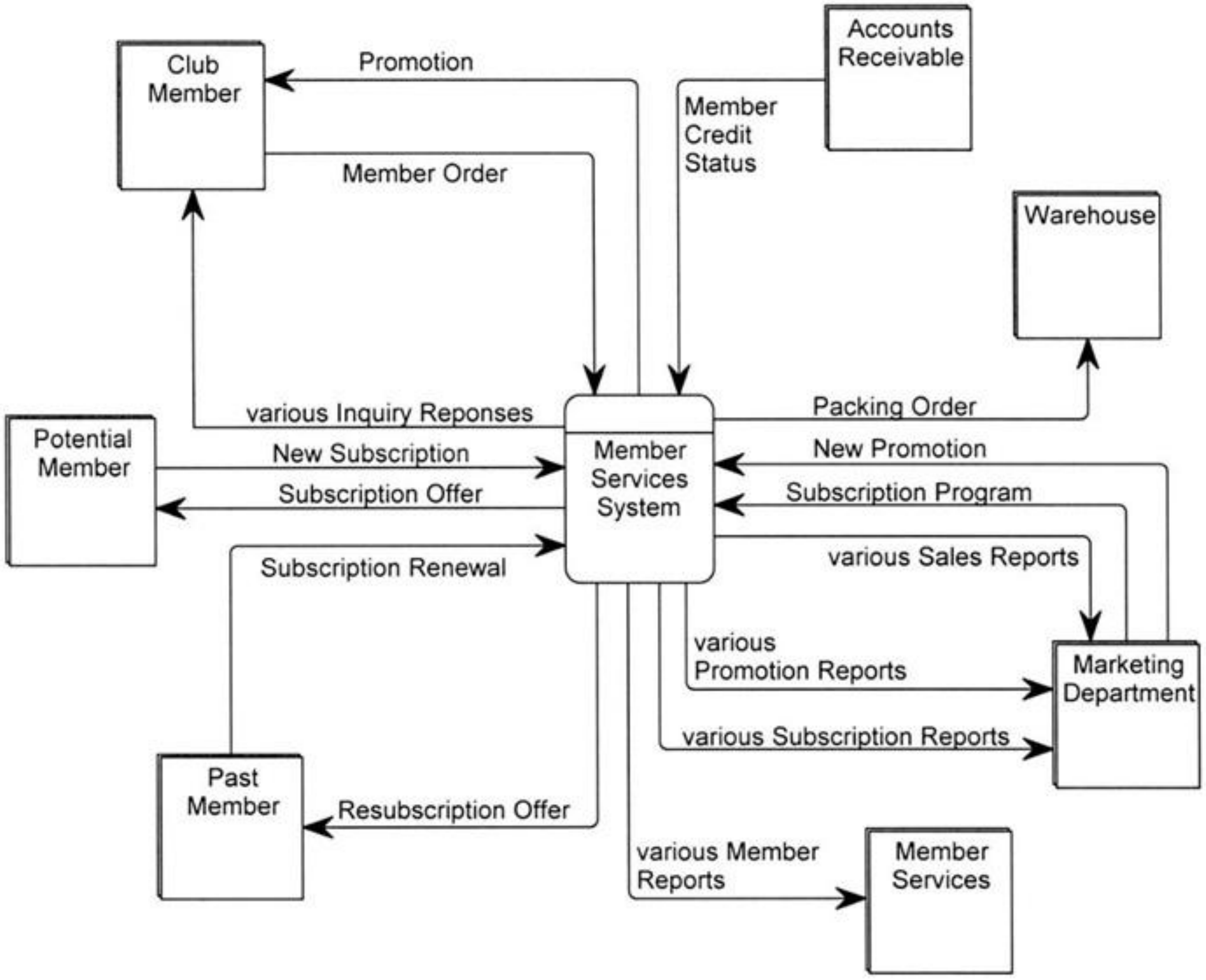
OK | Cancel | Spell | Delete

Text | Len: 1500

Context Data Flow Diagram

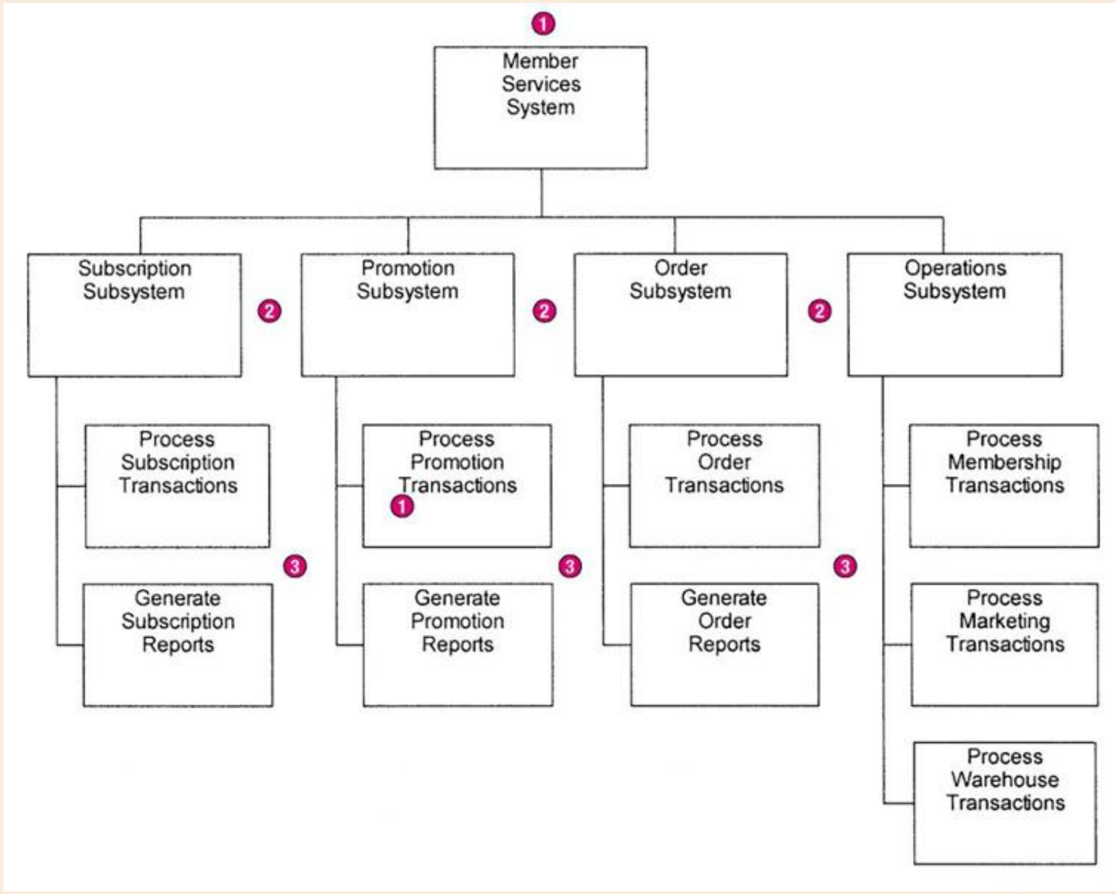
- **Context data flow diagram** - a process model used to document the scope for a system. Also called the environmental model.

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

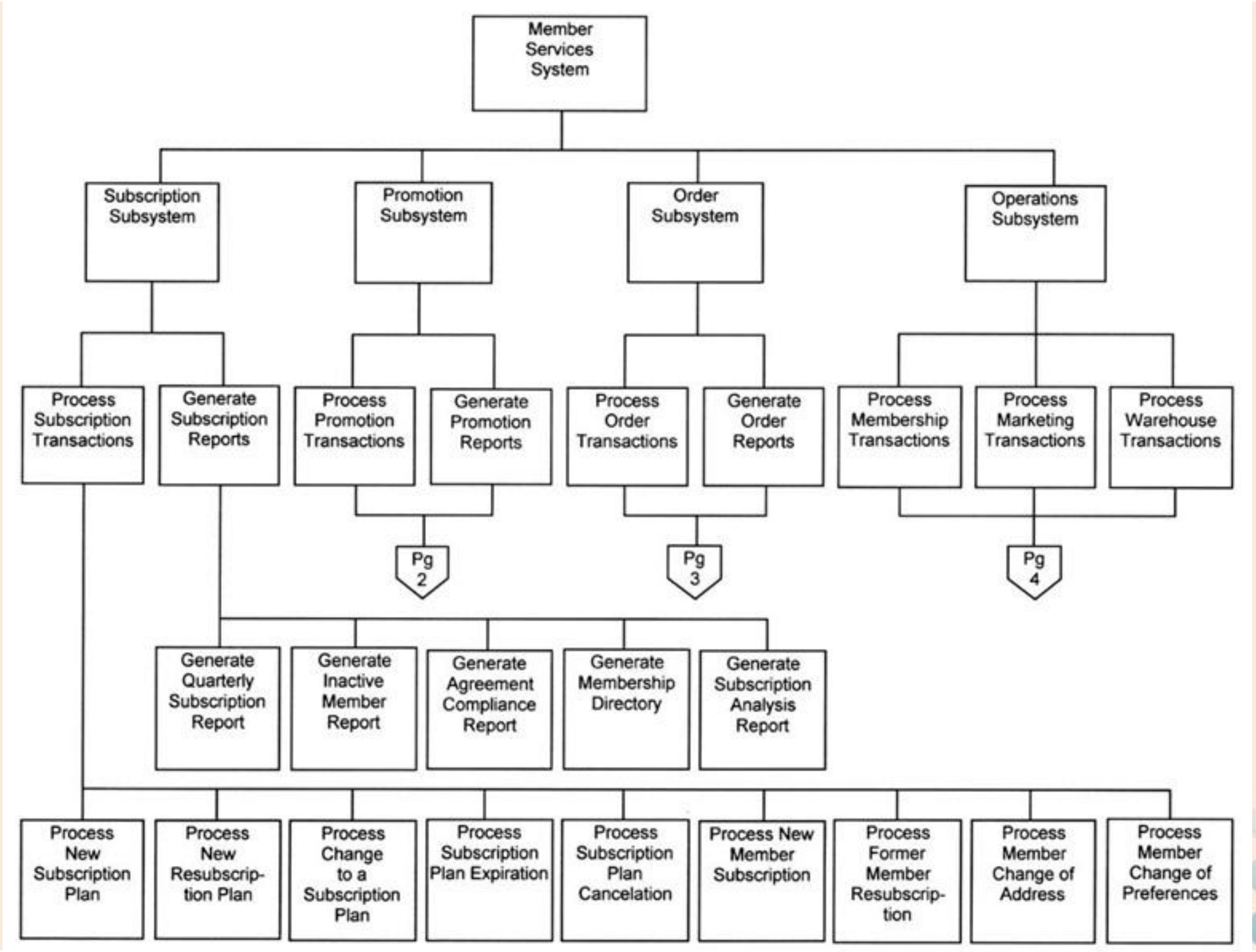
Events and Use Cases

- **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

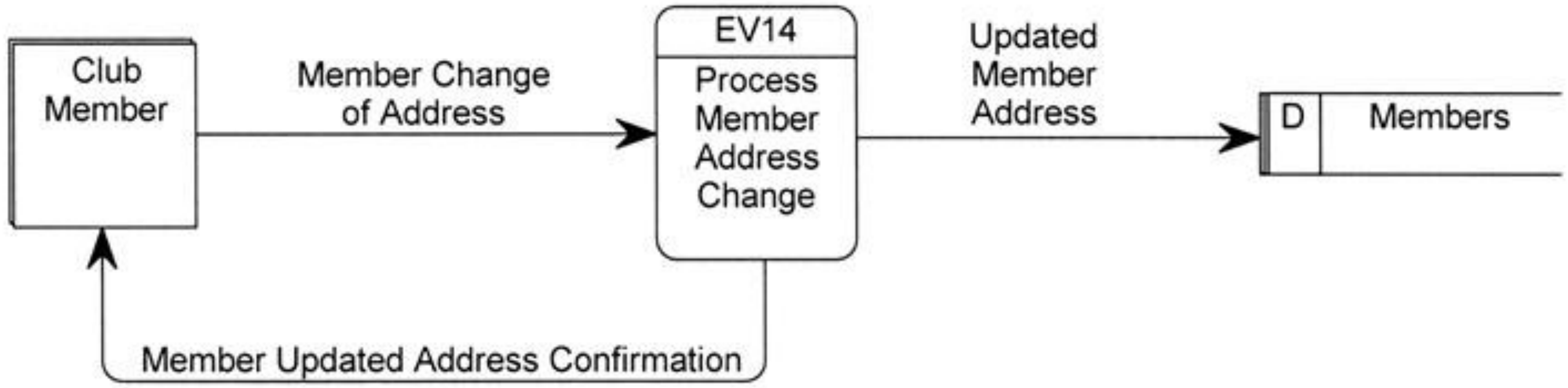

Member Services System
Subscription Subsystem
Promotion Subsystem
Order Subsystem
Operations Subsystem
Process Subscription Transactions
Generate Subscription Reports
Process Promotion Transactions
Generate Promotion Reports
Process Order Transactions
Generate Order Reports
Process Membership Transactions
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

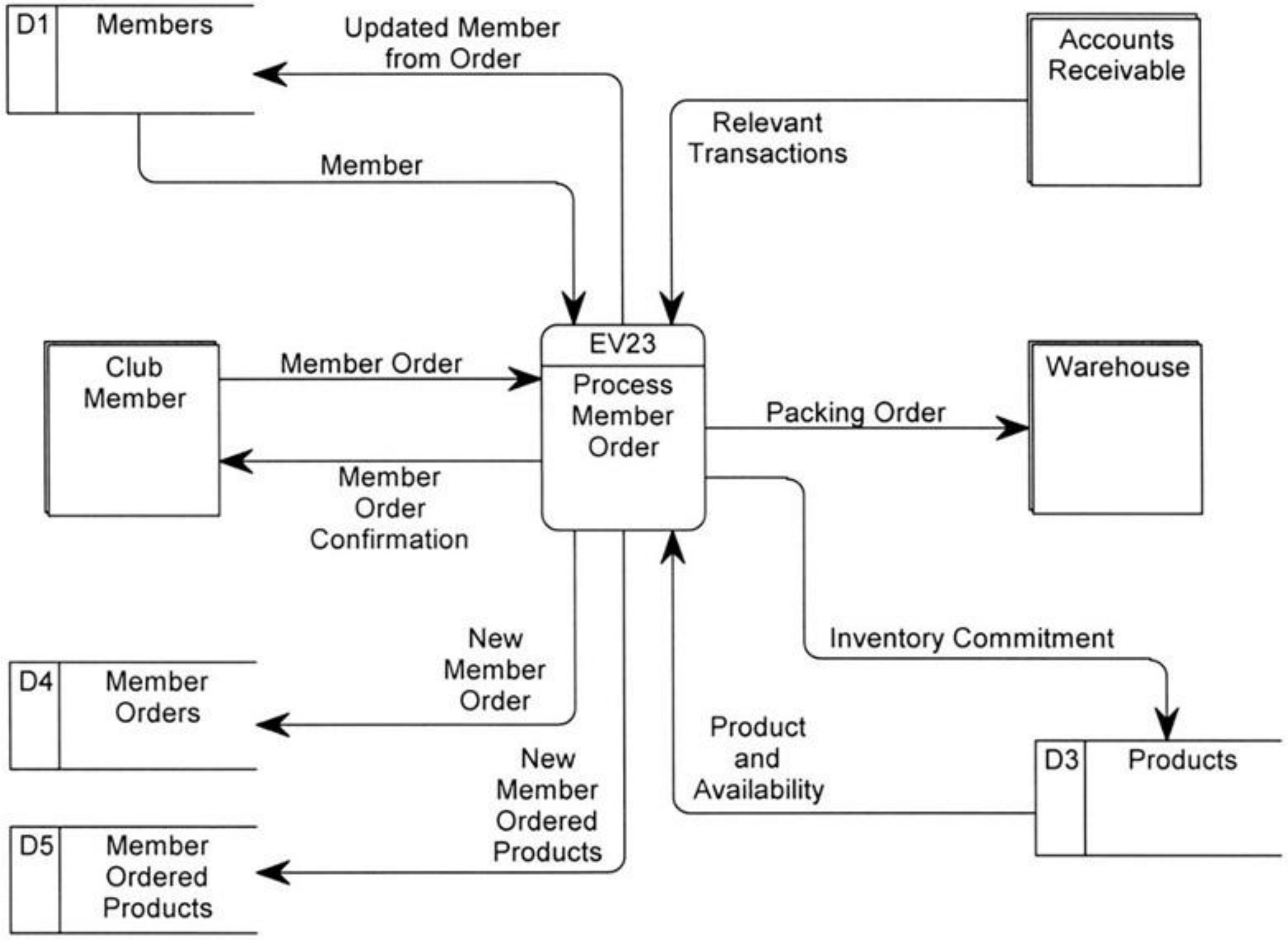
Event diagram – data flow diagram that depicts the context for a single event.

- 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.

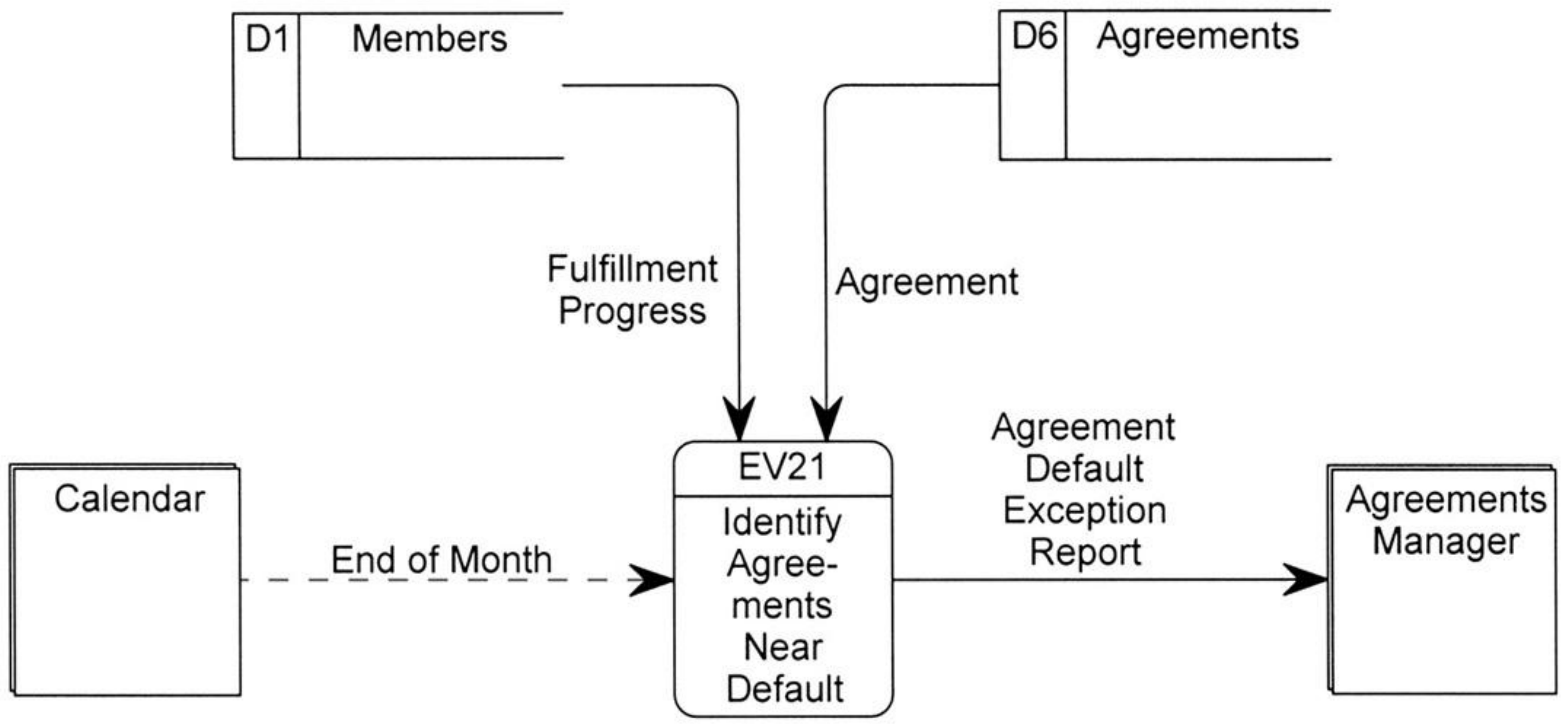
Simple Event Diagram



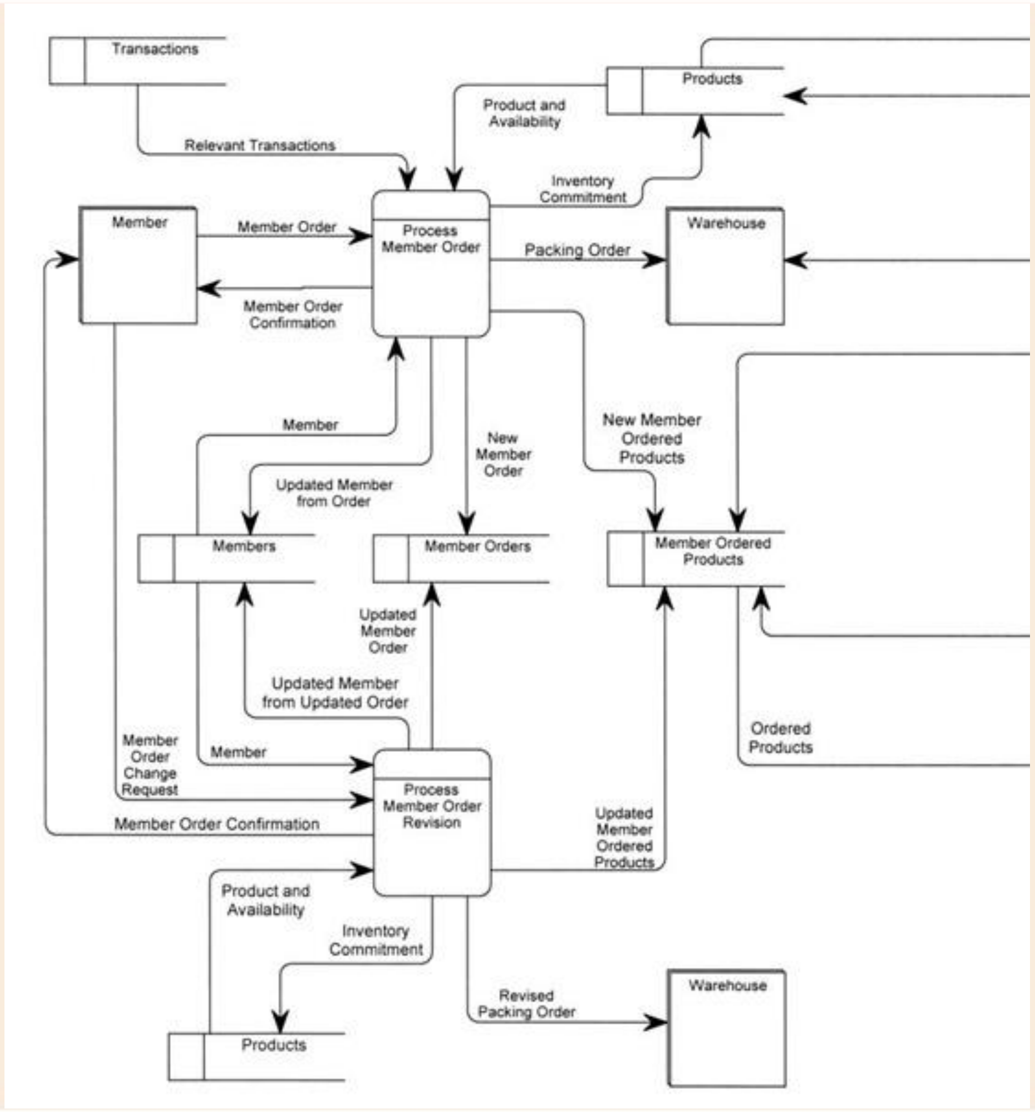
Event Diagram (more complex)



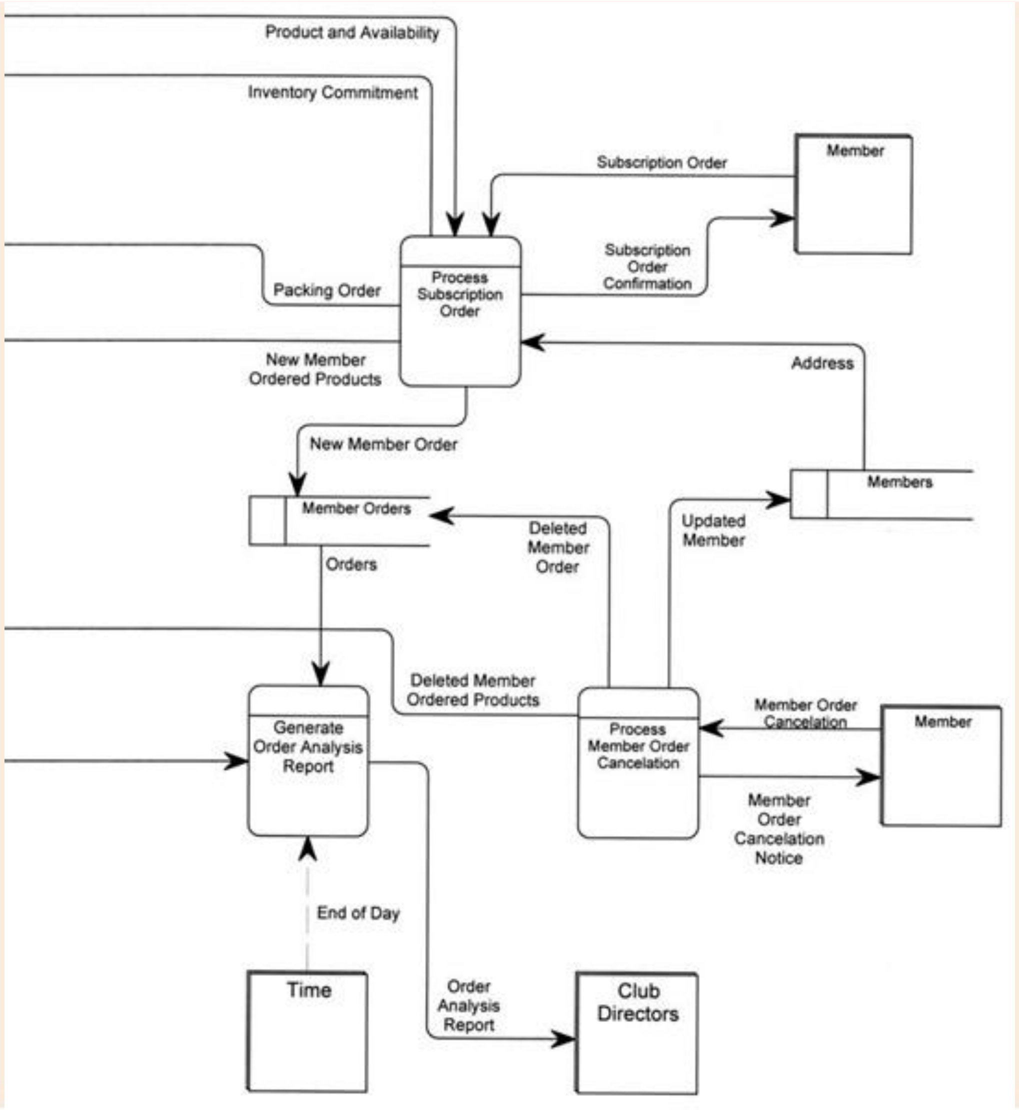
Temporal Event Diagram



System DFD


Transactions
Relevant Transactions
Products
Product and Availability
Inventory Commitment
Member
Member Order
Process Member Order
Member Order Confirmation
Warehouse
Packing Order
Member
Updated Member from Order
New Member Order
New Member Ordered Products
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

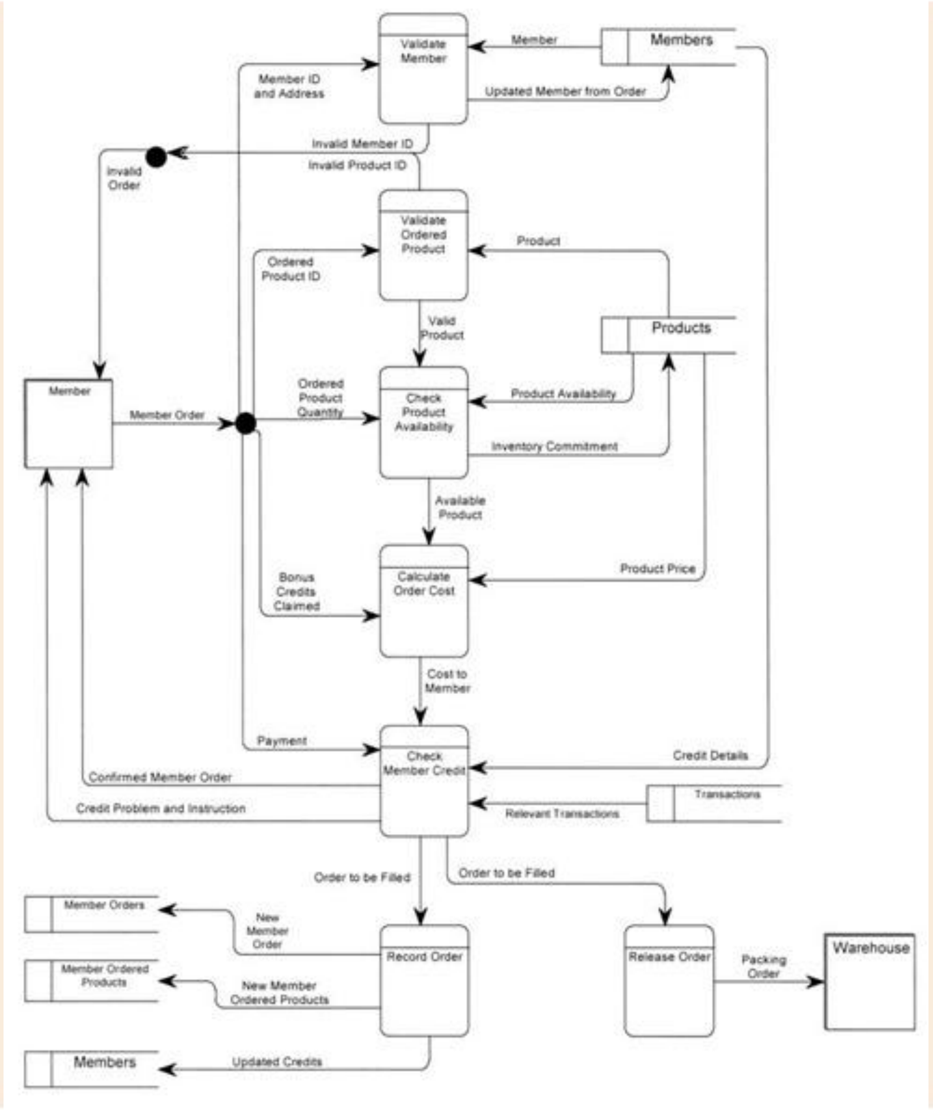
Balancing - a concept that requires that data flow diagrams at different levels of detail reflect consistency and completeness

- 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

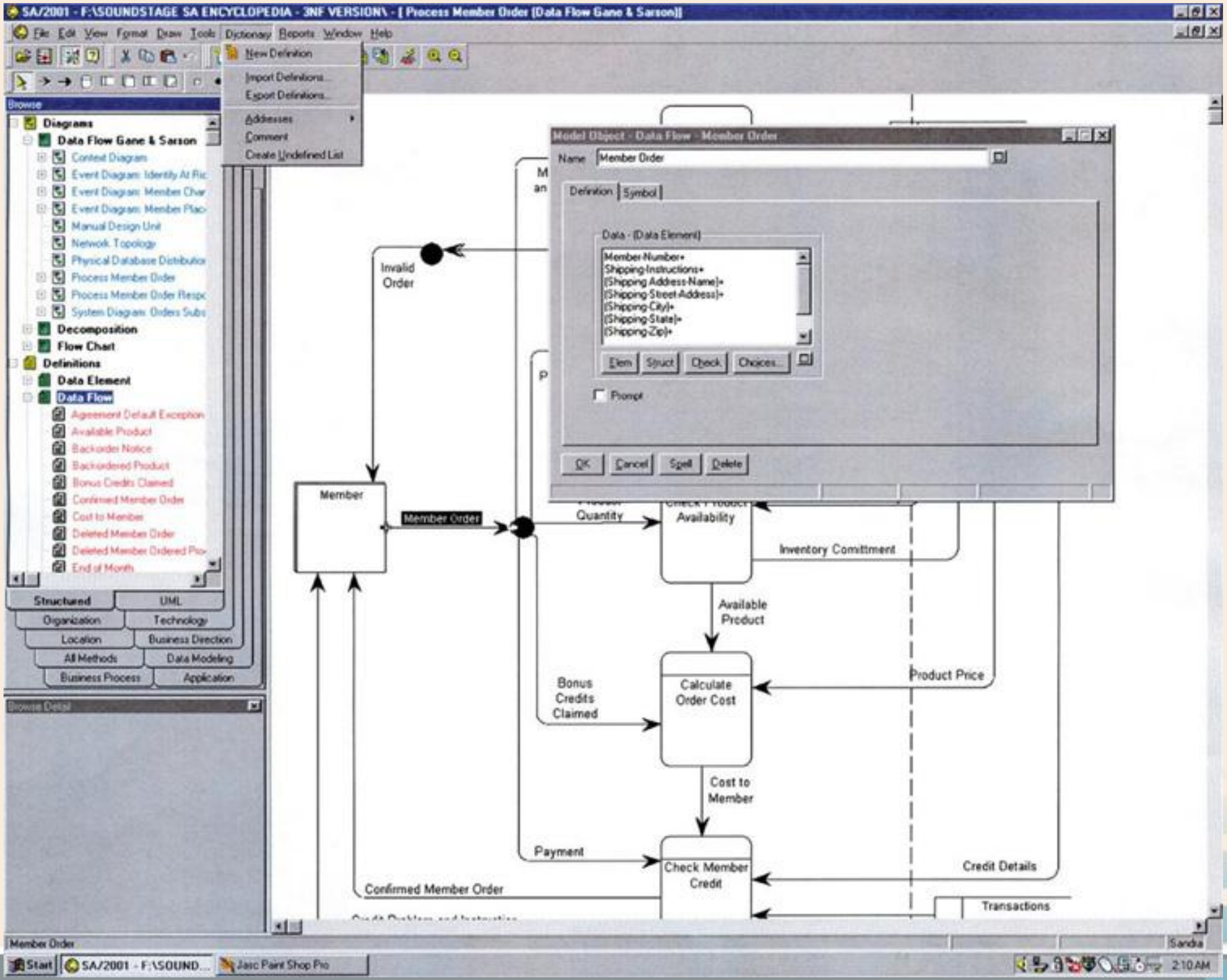
Primitive Diagrams

- 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

Primitive DFD
(see book for more readable copy)



Specifying a Data Flow Using a CASE Tool



Name: Member Order

Data - (Data Element)

Member-Number+
Shipping-Instructions+
(Shipping-Address-Name)+
(Shipping-Street-Address)+
(Shipping-City)+
(Shipping-State)+
(Shipping-Zip)+

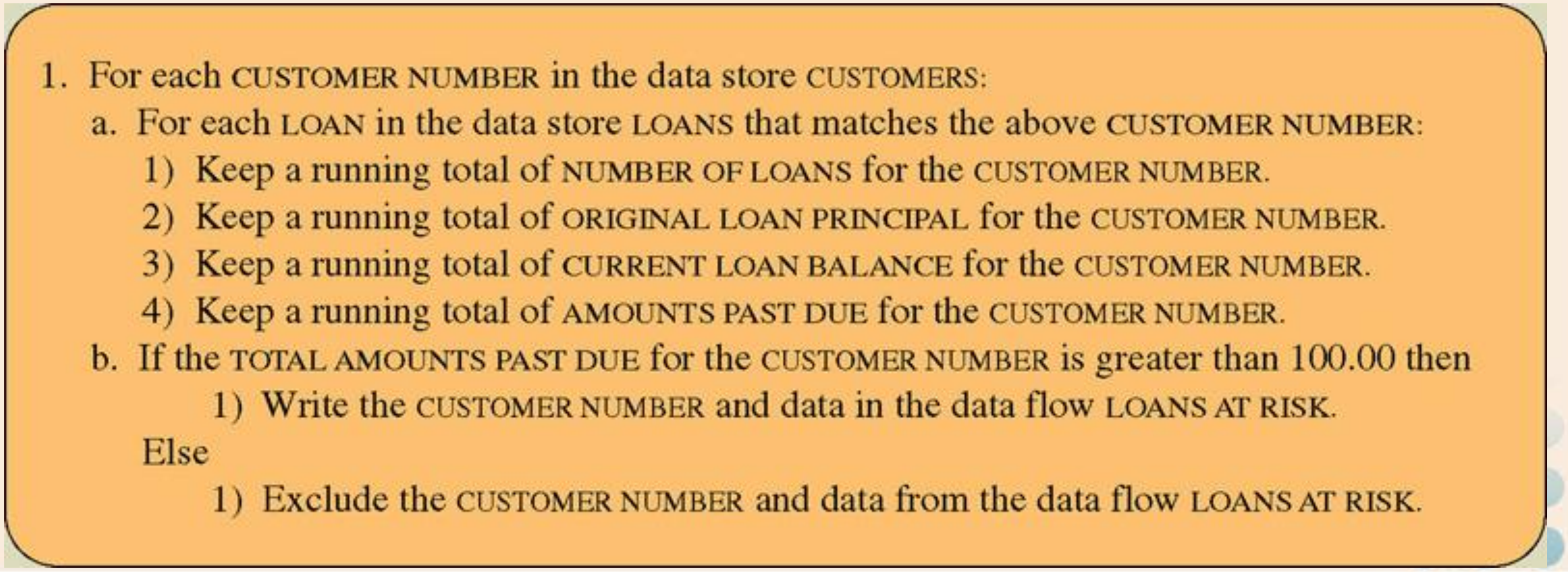
Process Logic

- Data Flow Diagrams good for identifying and describing processes
- Not good at showing logic inside processes
- Need to specify detailed instructions for elementary processes

Structured English

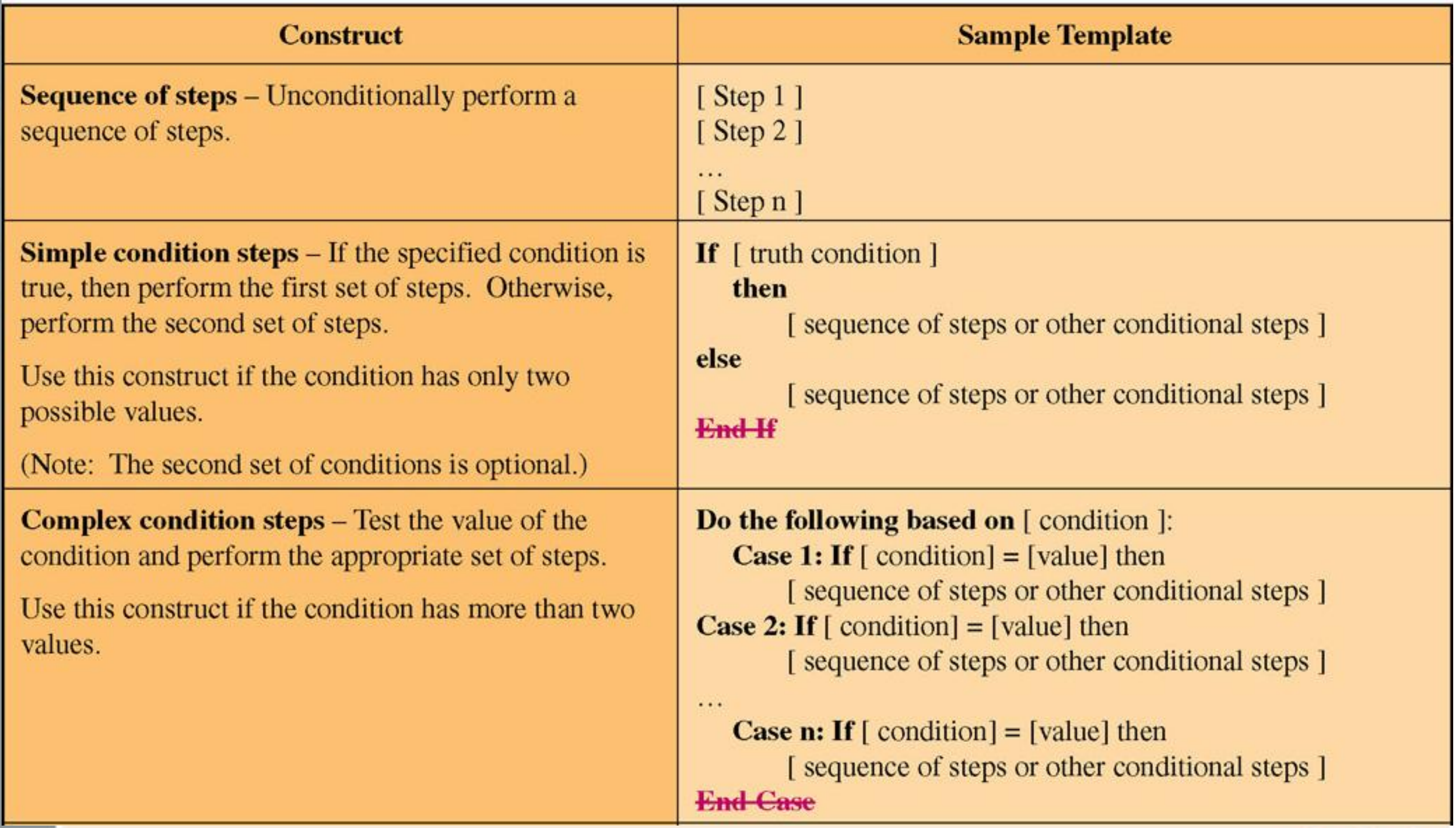
Structured English – a language syntax for specifying the logic of a process.

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



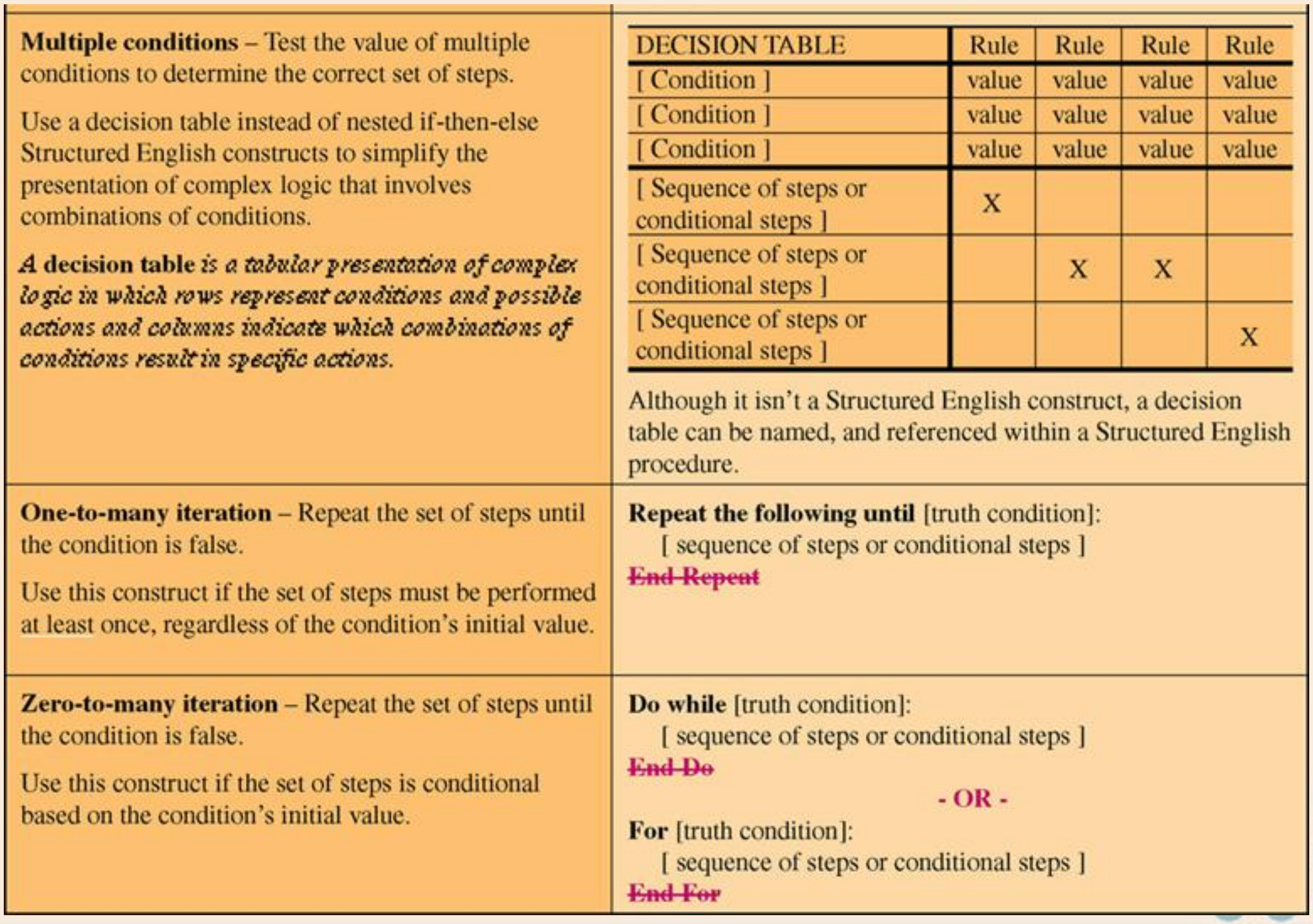
```
1. For each CUSTOMER NUMBER in the data store CUSTOMERS:
   a. For each LOAN in the data store LOANS that matches the above CUSTOMER NUMBER:
      1) Keep a running total of NUMBER OF LOANS for the CUSTOMER NUMBER.
      2) Keep a running total of ORIGINAL LOAN PRINCIPAL for the CUSTOMER NUMBER.
      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~~
```

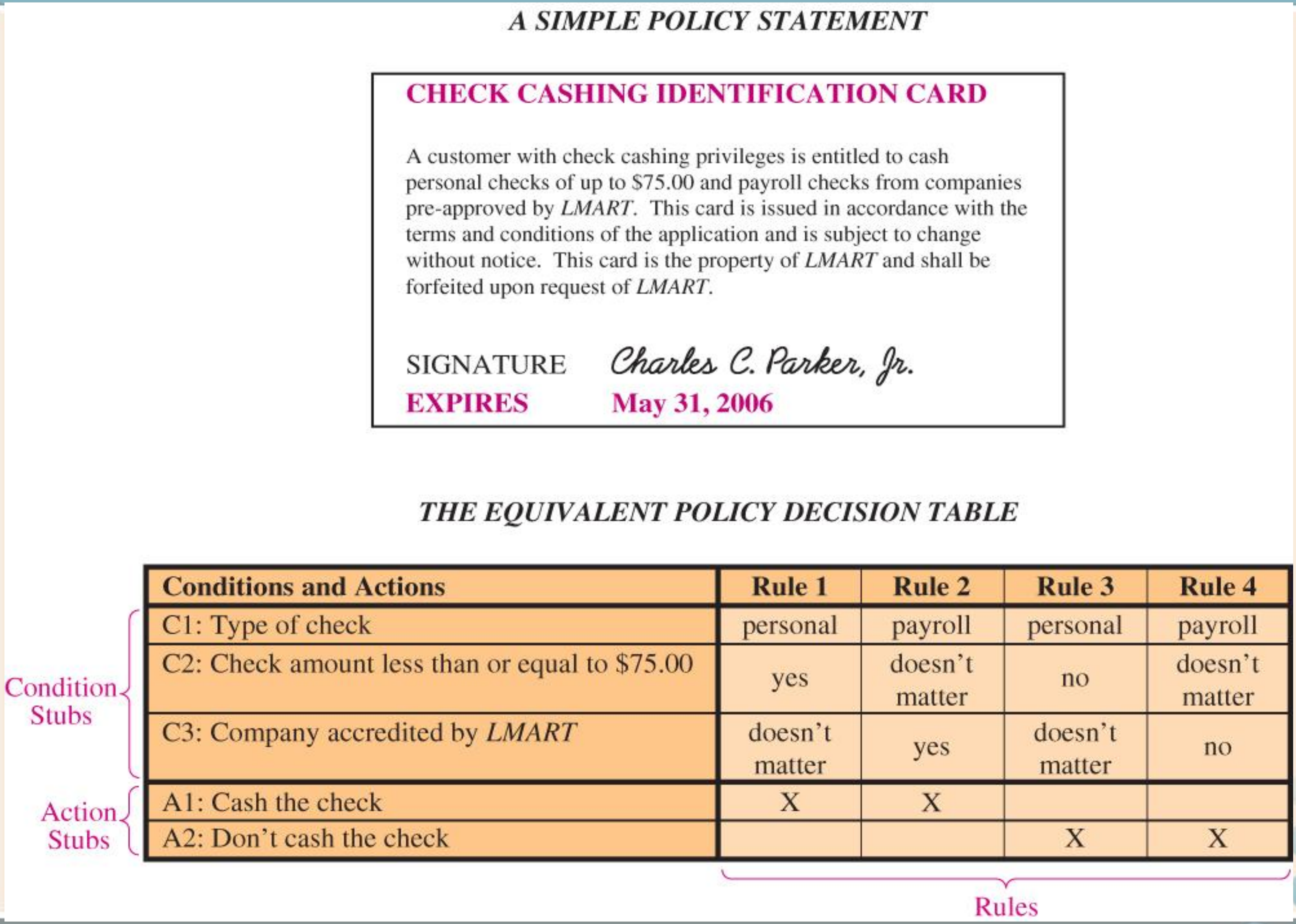
Policies and Decision Tables

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*.

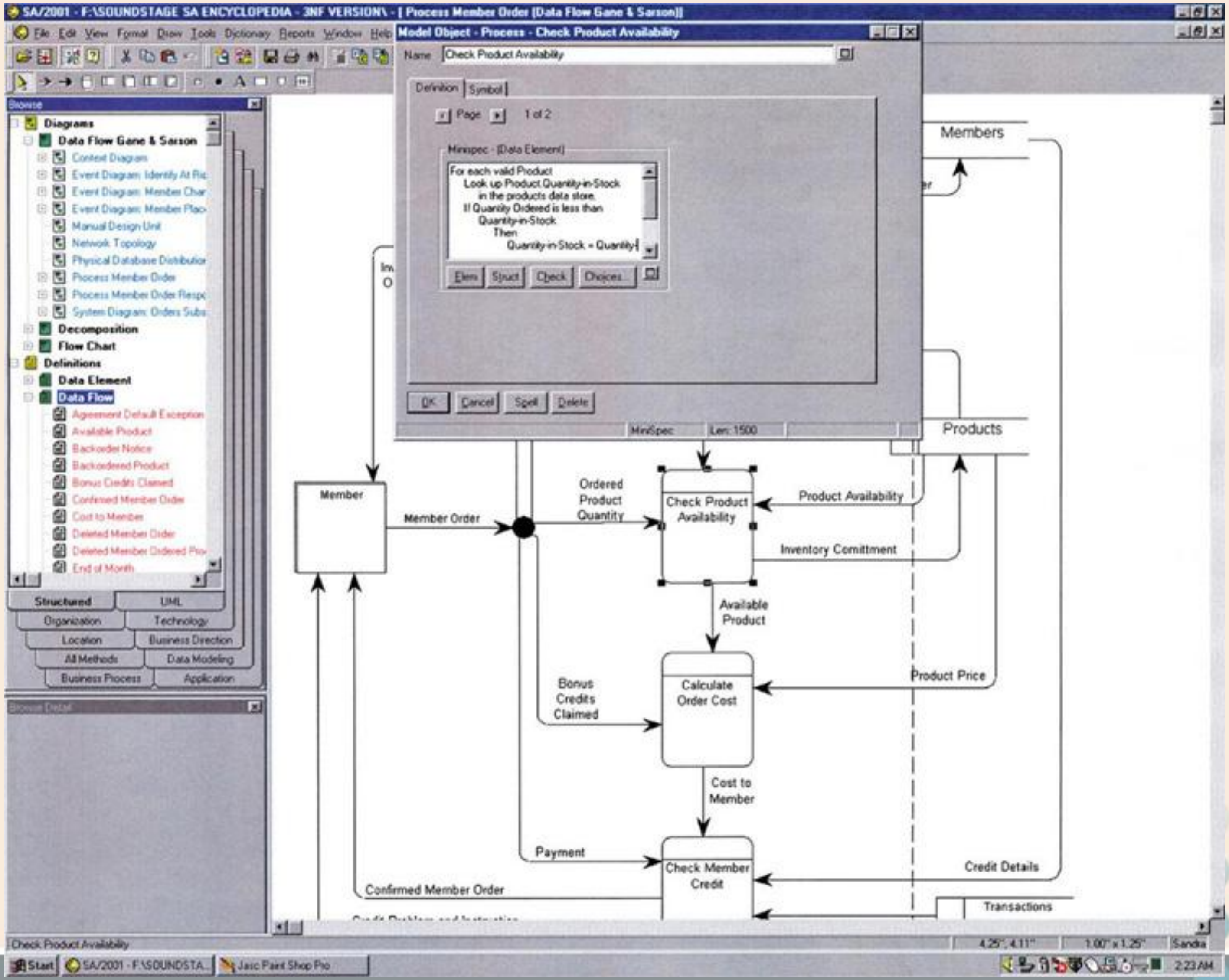
SIGNATURE Charles C. Parker, Jr.
EXPIRES **May 31, 2006**

THE EQUIVALENT POLICY DECISION TABLE

	Conditions and Actions	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**
Condition Stubs	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
Action Stubs	A1: Cash the check	X	X		
	A2: Don't cash the check			X	X

Rules

Describing an Elementary Process Using a CASE Tool



Name: Check Product Availability

Page 1 of 2

Minispec - (Data Element)

```
For each valid Product
    Look up Product Quantity-in-Stock
        in the products data store.
    If Quantity Ordered is less than
        Quantity-in-Stock
        Then
            Quantity-in-Stock = Quantity
```

MiniSpec Len: 1500

Check Product Availability 4.25", 4.11" 1.00" x 1.25" Sandra