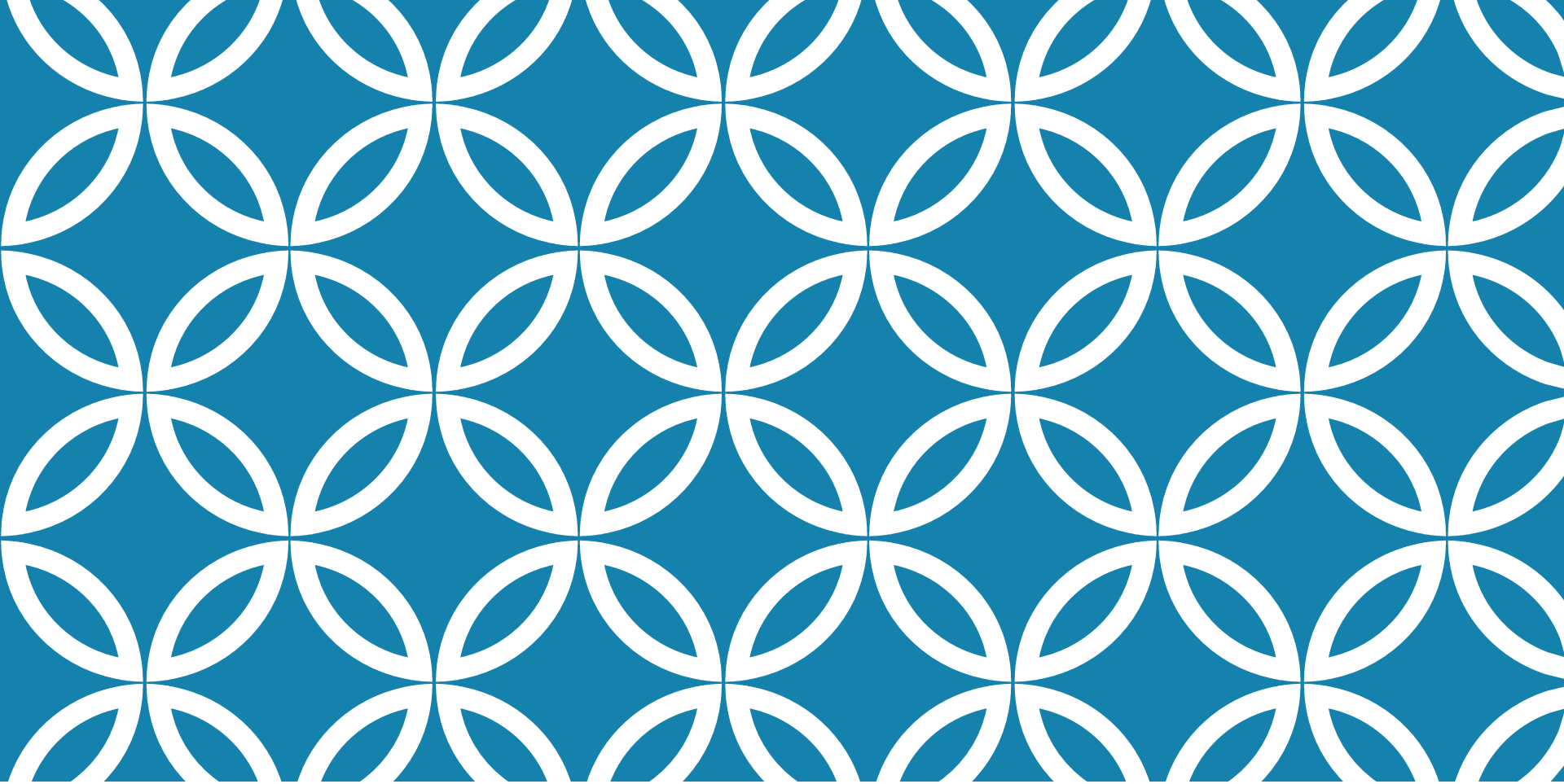


PROCESS MODELING

System Analysis & Design Course

Sharif University of Technology

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.

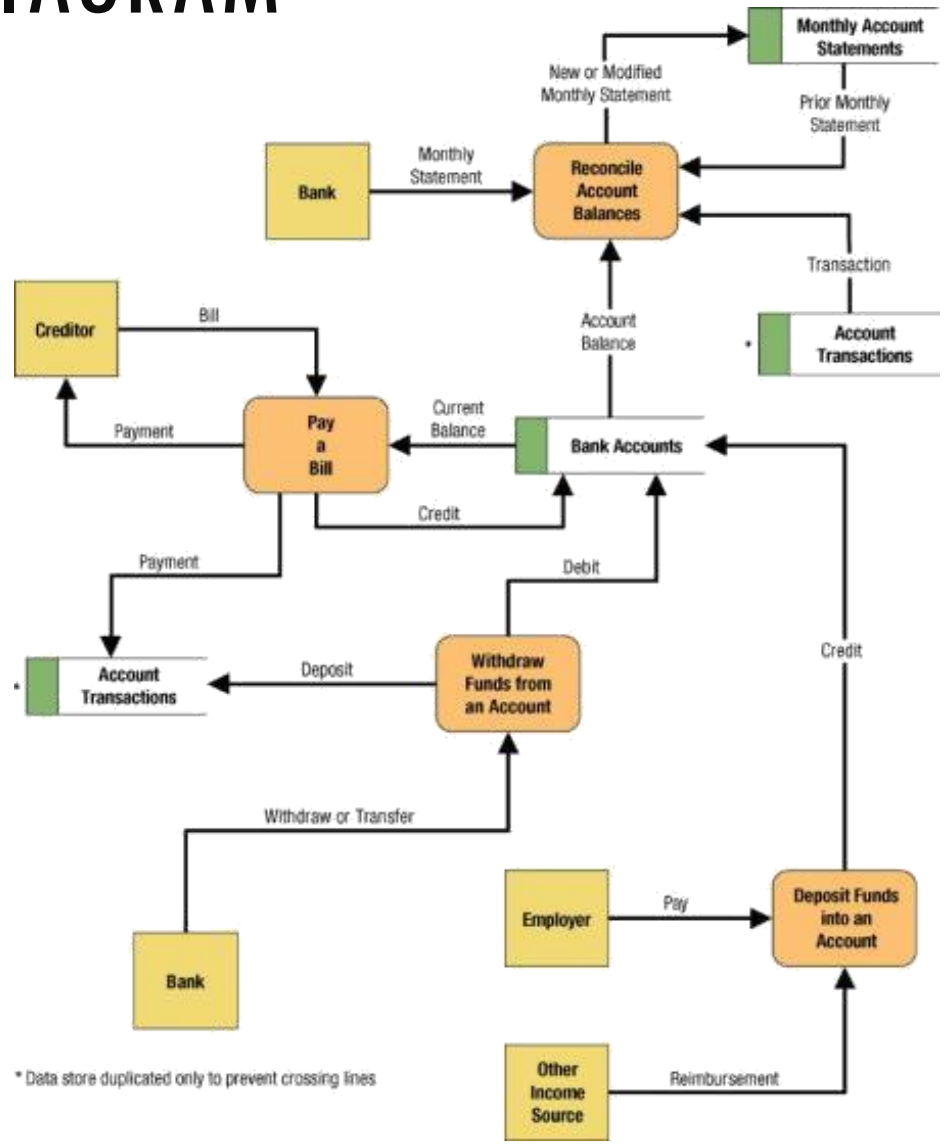
PROCESS MODELING

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

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

SIMPLE DATA FLOW DIAGRAM

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.



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

ACTIVITY DIAGRAM

Activity diagrams are used for

- documenting existing process
- analyzing new Process Concepts
- finding reengineering opportunities.

The diagrams describe the state of activities by showing the sequence of activities performed.

- they can show activities that are conditional or parallel.

ACTIVITY DIAGRAM CONCEPTS

An activity is trigged by one or more events and activity may result in one or more events that may trigger other activity or processes.

Events start from start symbol and end with finish marker having activities in between connected by events.

The activity diagram represents the decisions, iterations and parallel/random behavior of the processing.

- They capture actions performed.
- They stress on work performed in operations (methods).

WHEN TO USE ACTIVITY DIAGRAMS

The main reason to use activity diagrams is to model the workflow behind the system being designed.

Activity Diagrams are also useful for:

- analyzing a use case by describing what actions need to take place and when they should occur
- describing a complicated sequential algorithm
- modeling applications with parallel processes

Activity Diagrams should not take the place of interaction diagrams and state diagrams.

Activity diagrams do not give detail about how objects behave or how objects collaborate.

COMPONENTS

An *activity* is an ongoing, though interruptible, execution of a step in a workflow (such as an operation or transaction)

- Represented with a rounded rectangle.
- Text in the activity box should represent an activity (verb phrase in present tense).

COMPONENTS

An *event* is triggered by an activity. It specifies a significant occurrence that has a location in time and space.

- An instance of an event (trigger) results in the flow from one activity to another.
- These are represented by directed straight lines emerging from triggering activity and ending at activity to be triggered. Label text for events should represent event but not the data involved.

A *decision* may be shown by labeling multiple output transitions of an activity with different guard conditions.

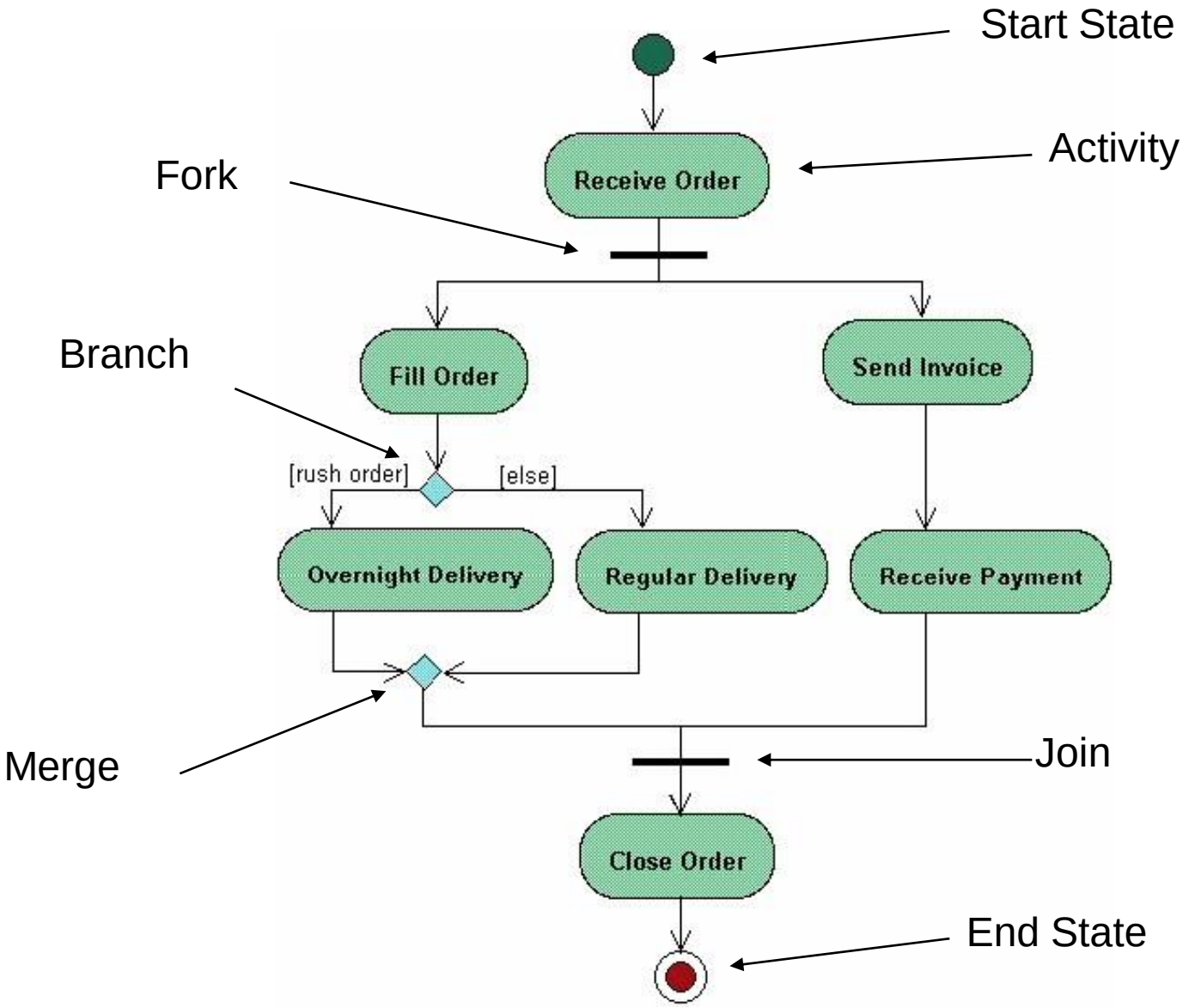
- For convenience a stereotype is provided for a decision: the traditional diamond shape, with one or more incoming arrows and with two or more outgoing arrows, each labeled by a distinct guard condition with no event trigger.

HOW TO DRAW AN ACTIVITY DIAGRAM

Diagrams are read from top to bottom and have branches and forks to describe conditions and parallel activities.

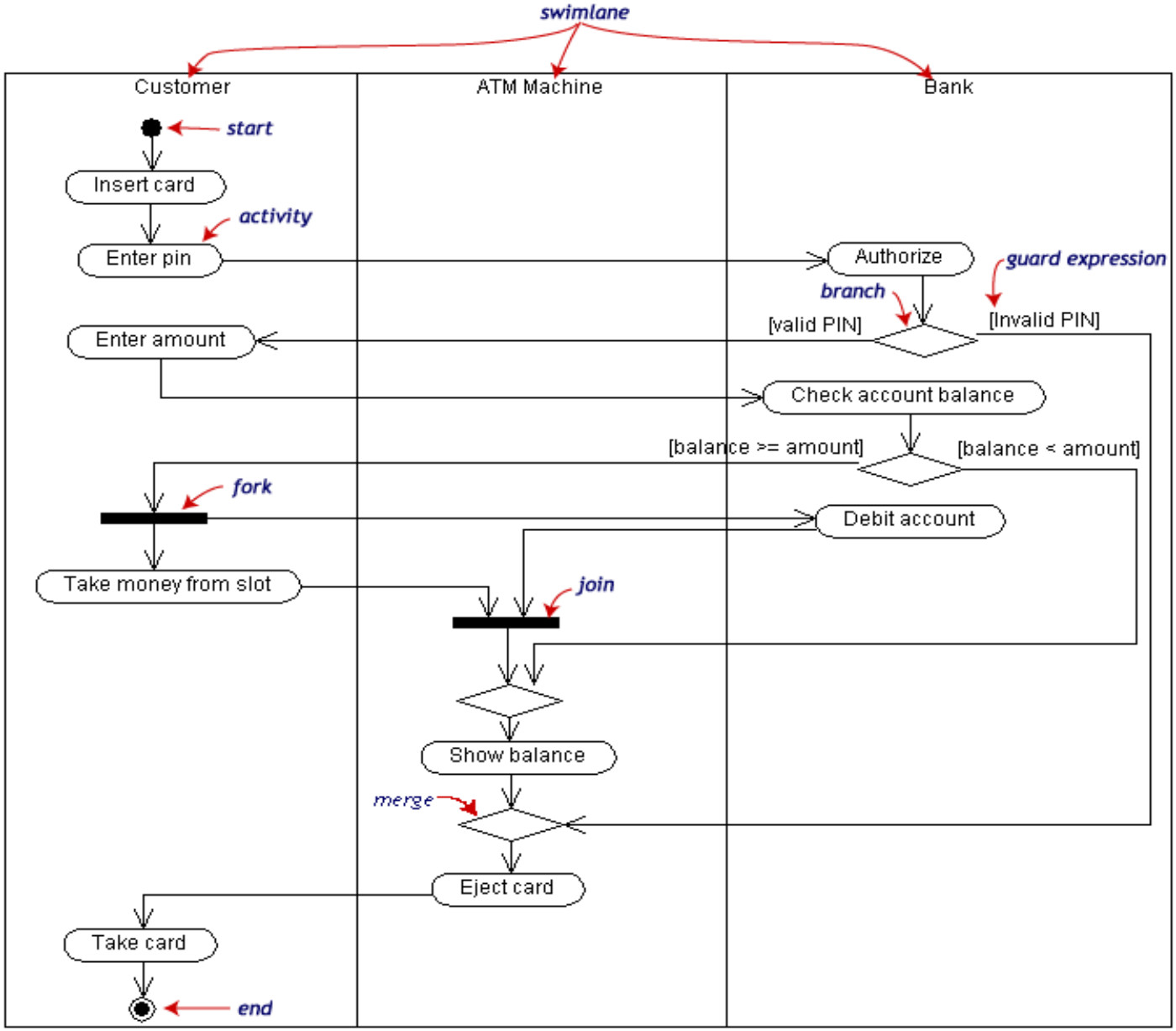
- A fork is used when multiple activities are occurring at the same time.
- A branch describes what activities will take place based on a set of conditions.
- All branches at some point are followed by a merge to indicate the end of the conditional behavior started by that branch.
- After the merge all of the parallel activities must be combined by a join before transitioning into the final activity state.

ACTIVITY DIAGRAM EXAMPLE

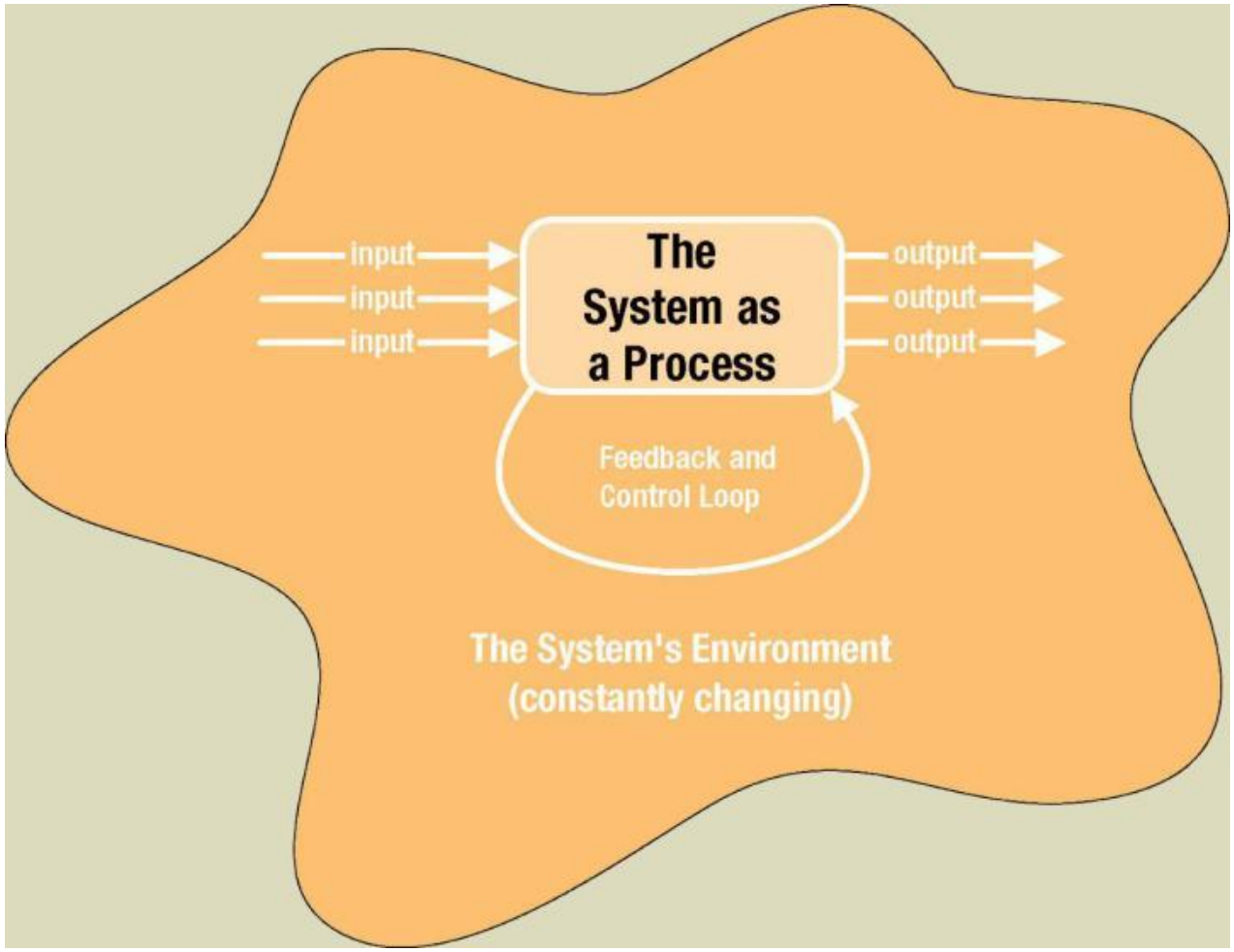


USE CASE

Withdraw money from a bank account through an ATM

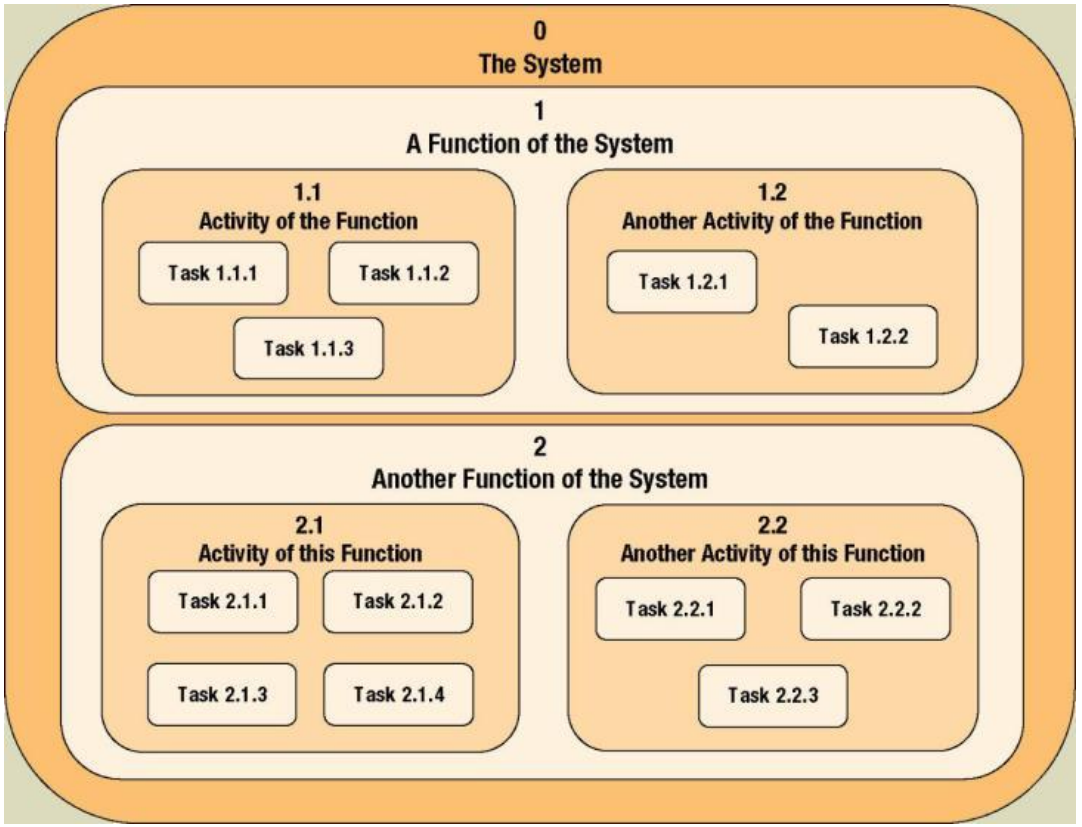

swimlane
Customer
ATM Machine
Bank
start
Insert card
activity
Enter pin
Authorize
guard expression
branch
[valid PIN]
[Invalid PIN]
Enter amount
Check account balance
[balance >= amount]
[balance < amount]
fork
Debit account
Take money from slot
join
Show balance
merge
Eject card
Take card
end

THE SYSTEM IS ITSELF A PROCESS



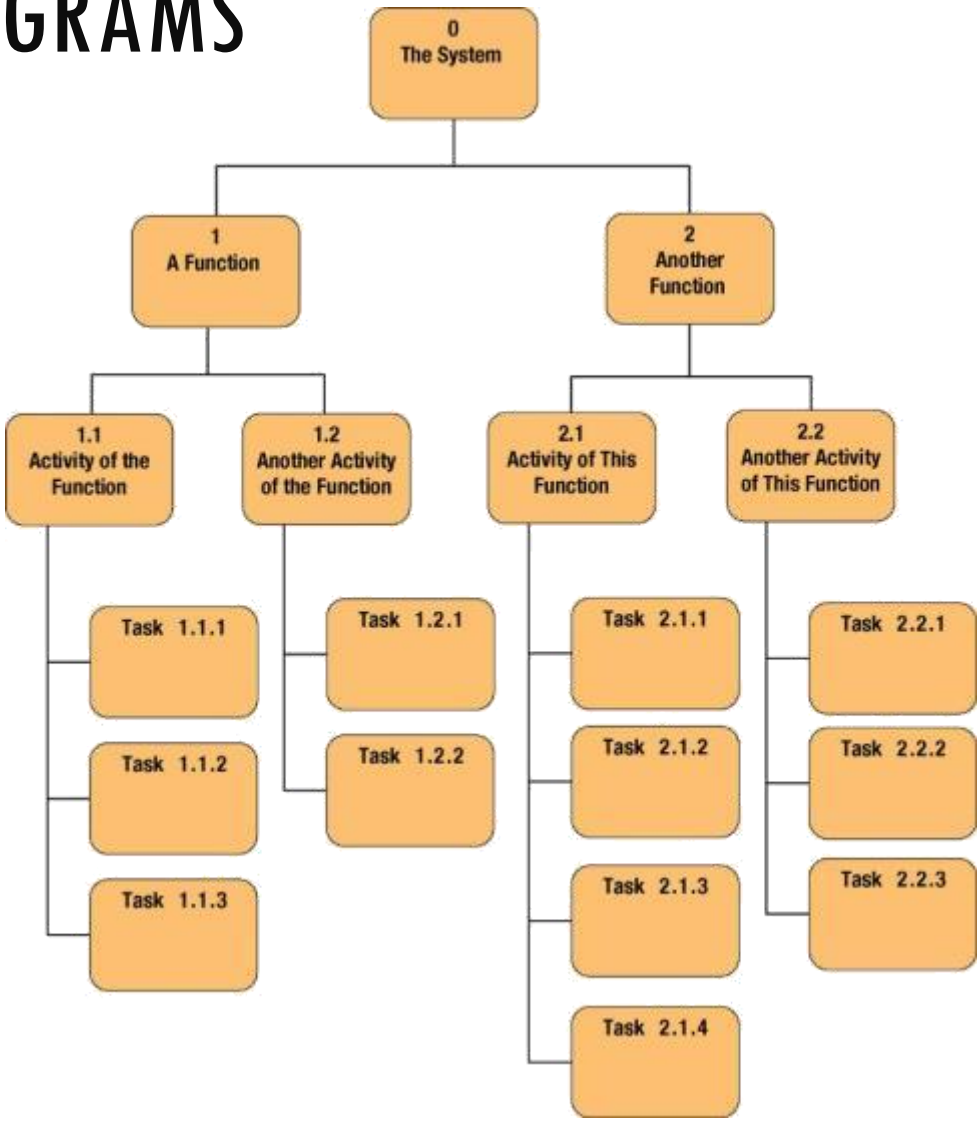
PROCESS DECOMPOSITION

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.

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.

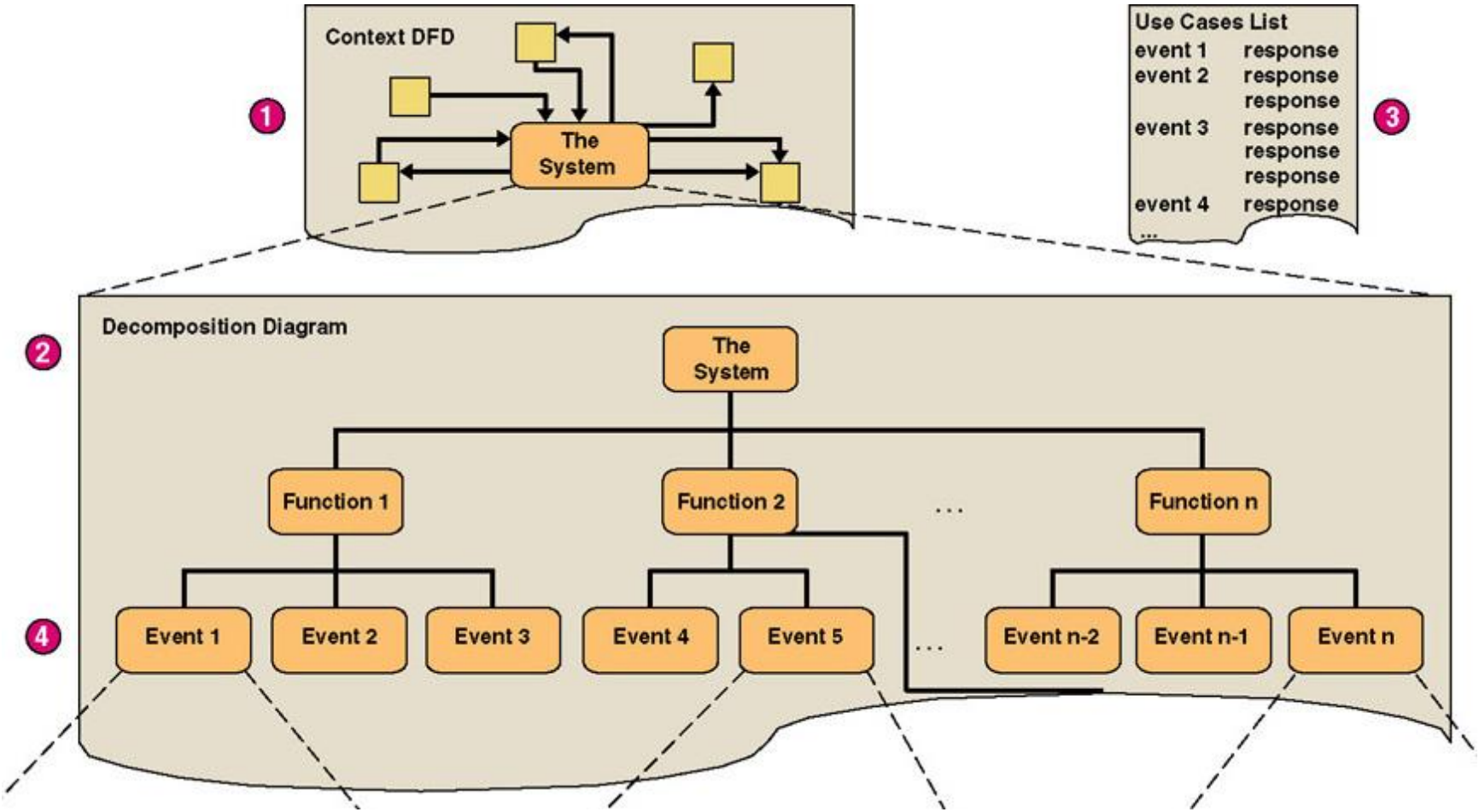
Systems analysis (primary focus of this course)

- Model existing system including its limitations
- Model target system's logical requirements
- Model candidate technical solutions
- Model the target technical solution

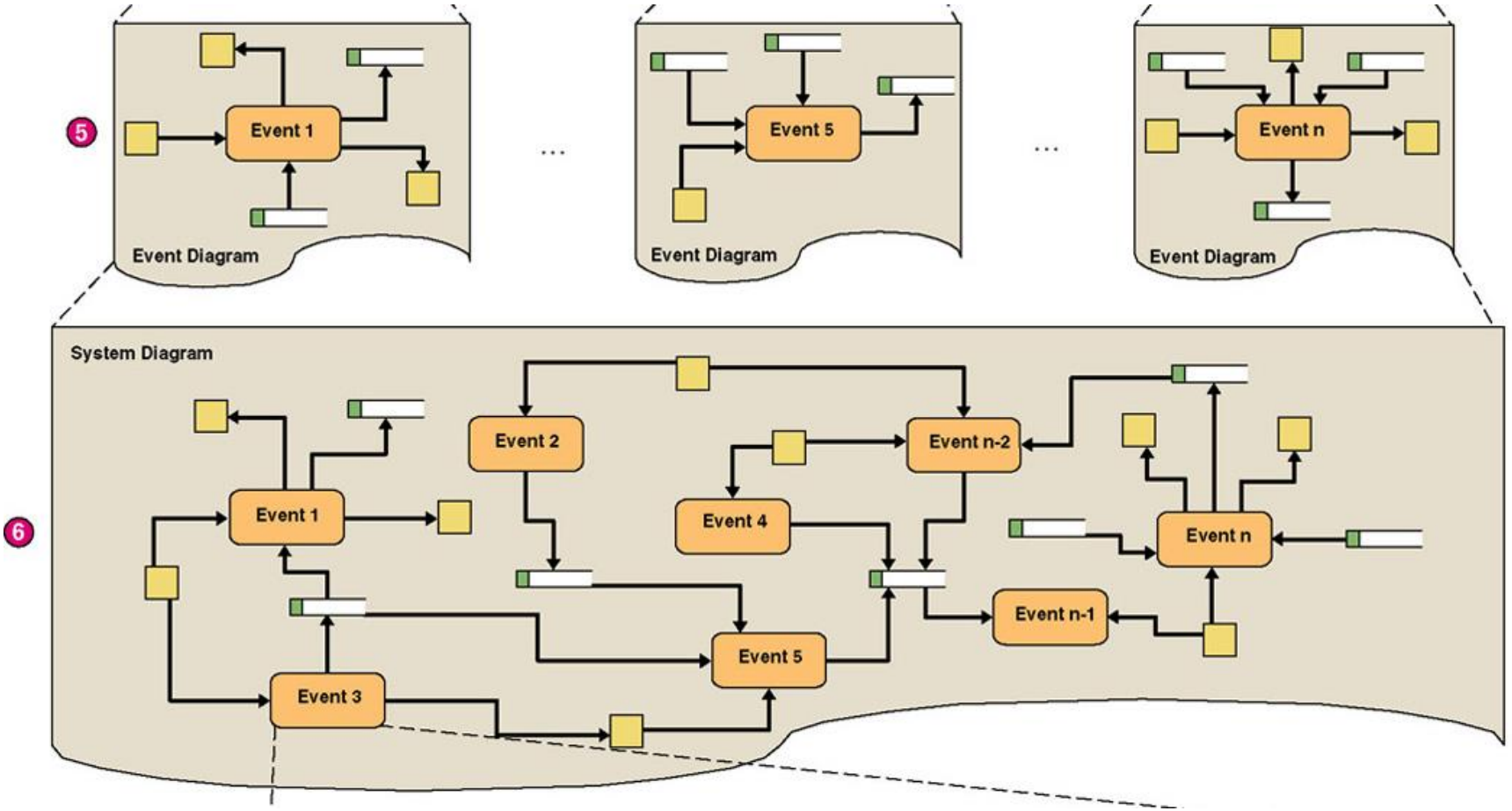
MODERN STRUCTURED ANALYSIS (MORE COMMONLY PRACTICED)

1. Draw context diagram 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 diagram (or event handler) for each event.
5. Merge event diagrams into a system diagram (or, for larger systems, subsystem diagrams).
6. Draw detailed, primitive diagrams for the more complex event handlers.
7. Document data flows and processes in data dictionary.

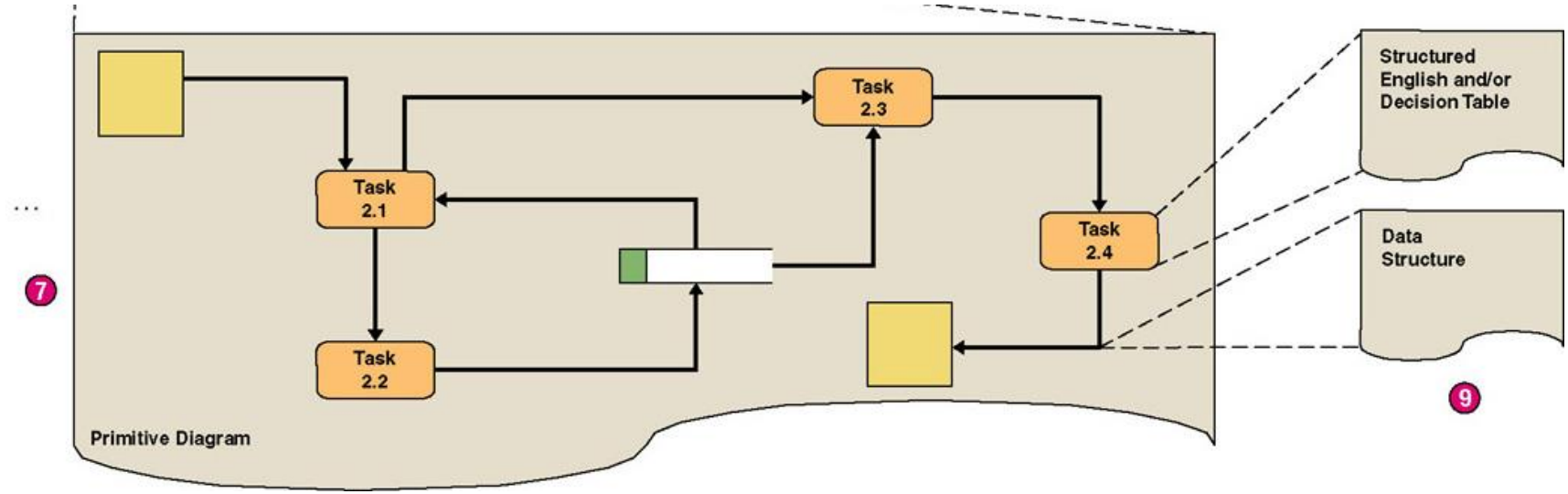
STRUCTURED ANALYSIS DIAGRAM PROGRESSION (1 OF 3)



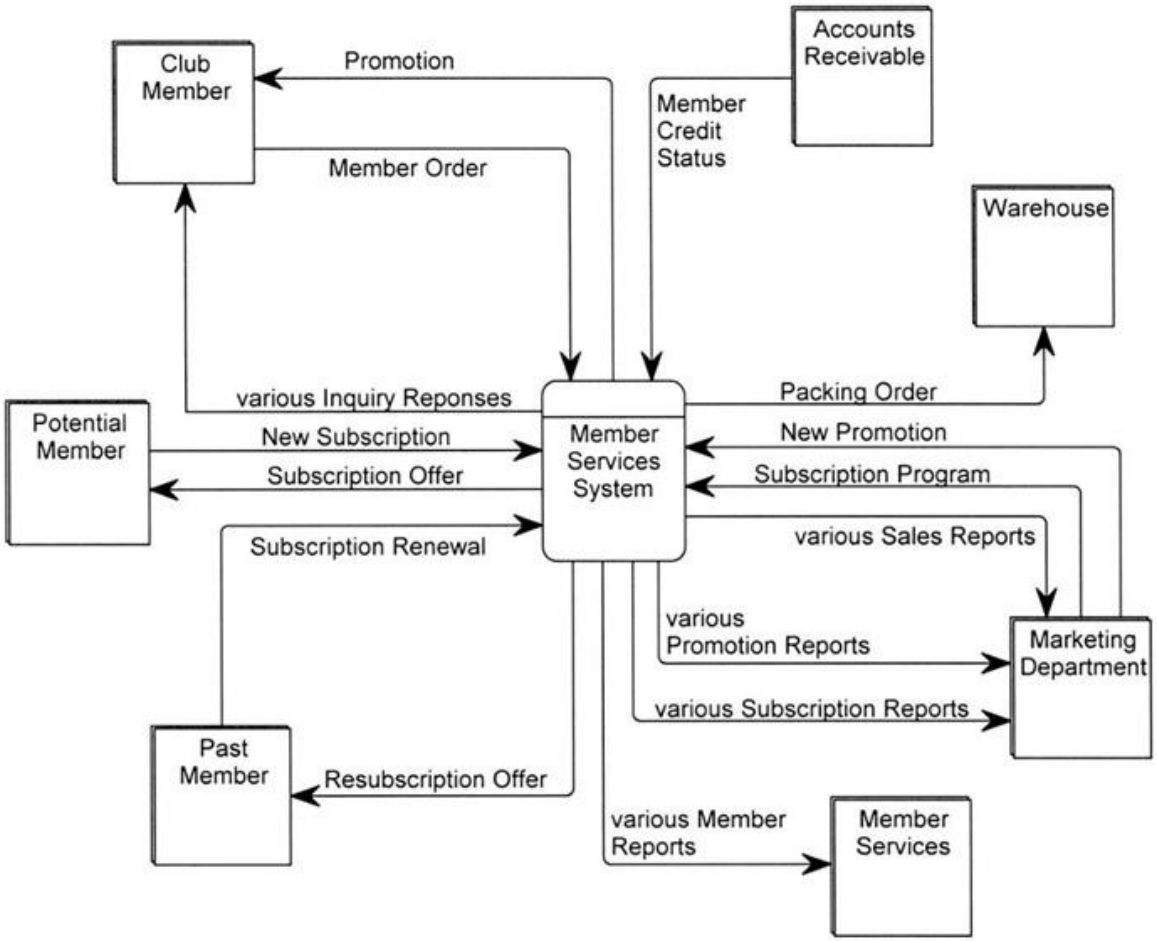
STRUCTURED ANALYSIS DIAGRAM PROGRESSION (2 OF 3)



STRUCTURED ANALYSIS DIAGRAM PROGRESSION (3 OF 3)



SOUNDSTAGE CONTEXT DFD

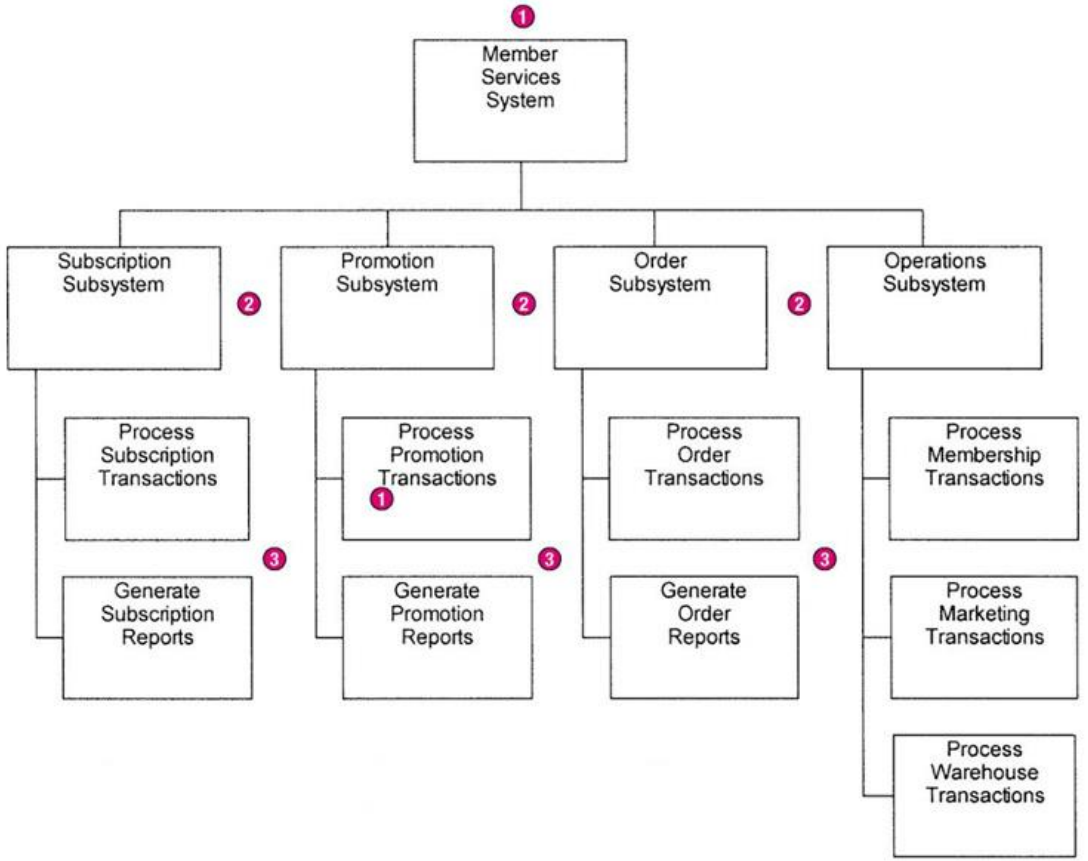


FUNCTIONAL DECOMPOSITION DIAGRAM

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.



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.

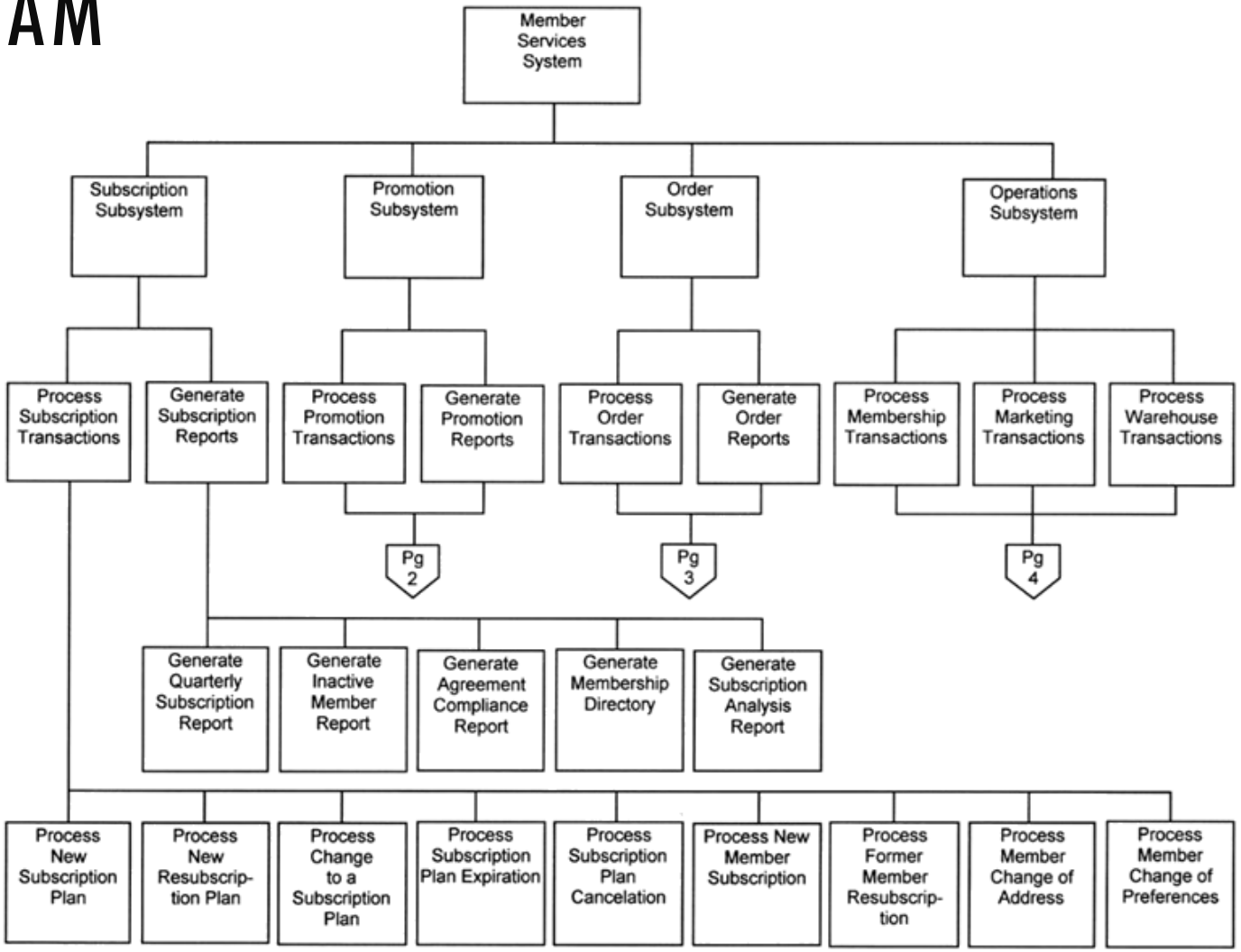
Use case – an analysis tool for finding and identifying business events and responses.

Actor – anything that interacts with a system.

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.

PARTIAL EVENT DECOMPOSITION DIAGRAM



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 data flows and data stores

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.

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

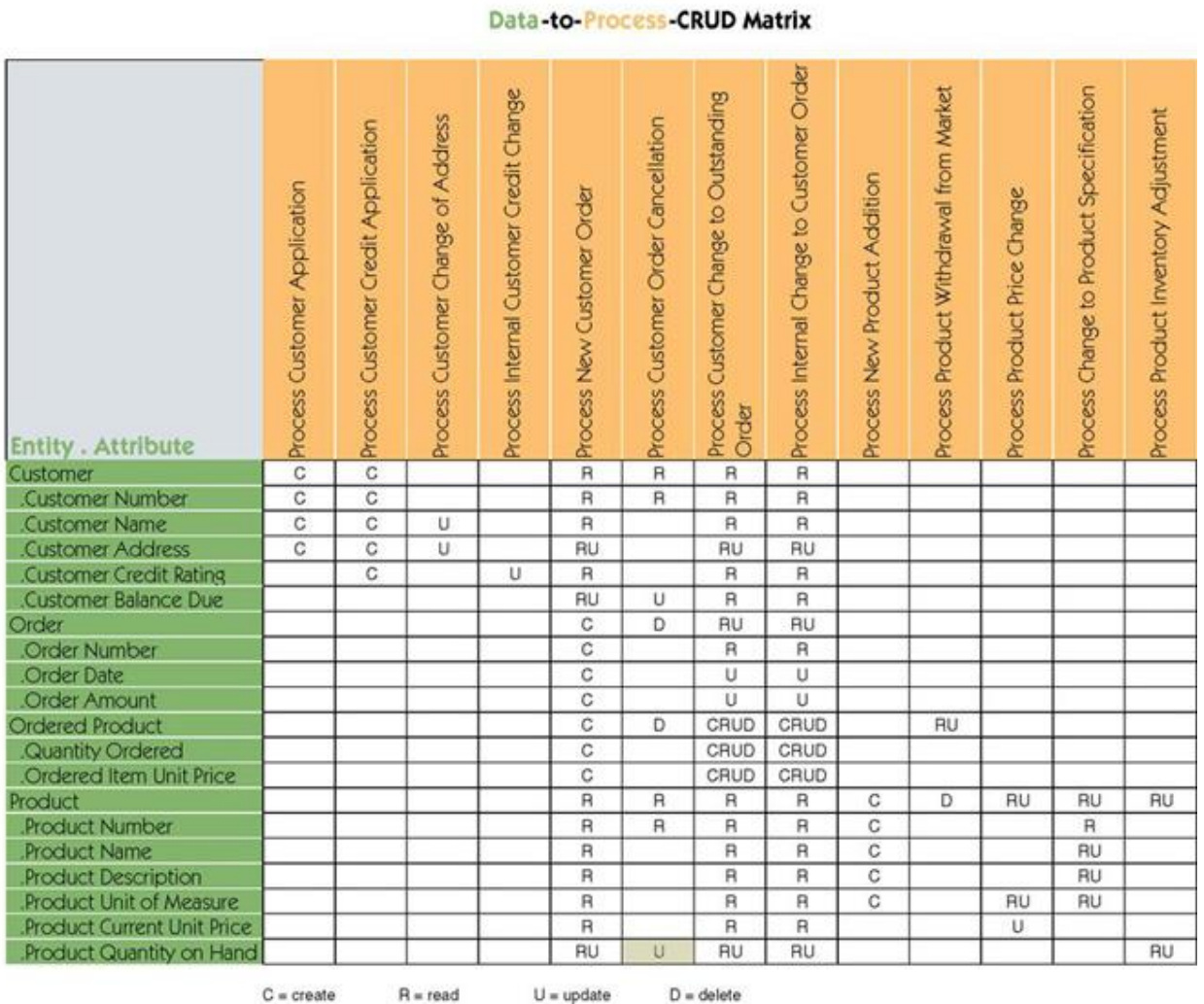
A customer with check cashing privileges is entitled to cash personal checks of up to $75.00 and payroll checks from companies pre-approved by *LMART*. This card is issued in accordance with the terms and conditions of the application and is subject to change without notice. This card is the property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

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