Chapter 3:
System Analysis

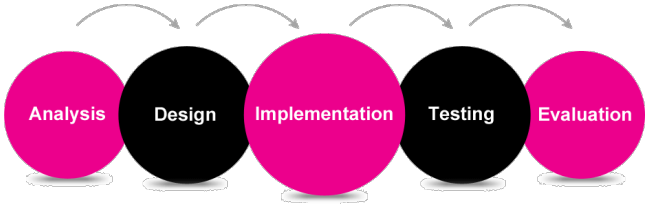
Prepared by: nursyuhada@puo.edu.my

Learning outcomes

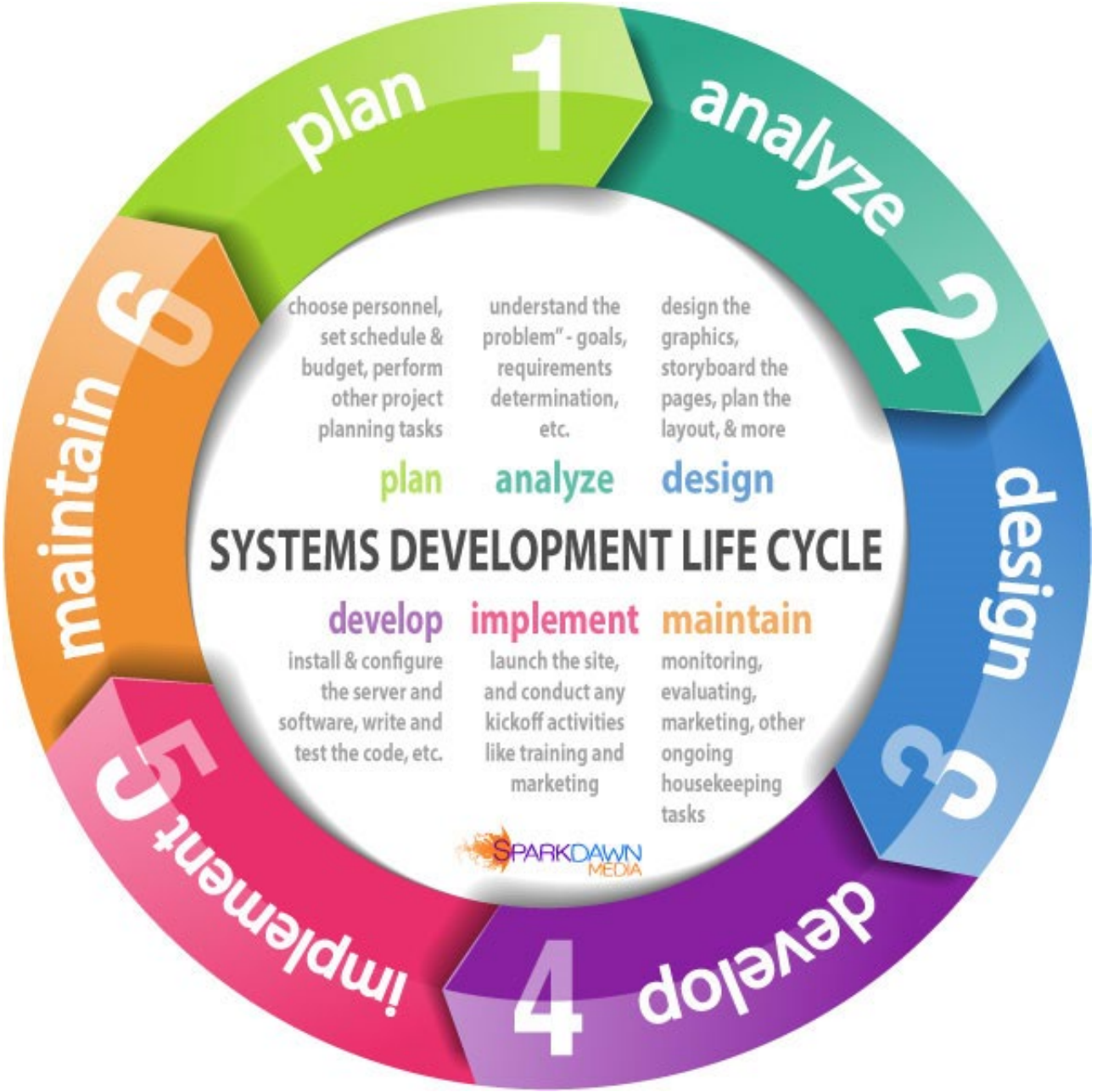
3.1 Conduct Preliminary analysis.

- Identify the system requirements
- Explain the technique used in fact finding
- Fact finding technique
- Analyze information
- Functional Decomposition Diagram (FDD)
- Unified Modelling Language (UML) Diagrams.

Introduction



- Systems analysis is the second of five phases in the systems development life cycle.
- The system Analysis Phase is the process of gathering facts about a systems project, preparing documentation, and creating models that will be used to design and develop the system.
- The Analysis Phase is to identify the overall direction that the project will take through the creation of the project strategy documents.


plan 1
analyze 2
design 3
develop 4
implement 5
maintain 6
choose personnel, set schedule & budget, perform other project planning tasks
understand the problem" - goals, requirements determination, etc.
design the graphics, storyboard the pages, plan the layout, & more
plan
analyze
design
SYSTEMS DEVELOPMENT LIFE CYCLE
develop
install & configure the server and software, write and test the code, etc.
implement
launch the site, and conduct any kickoff activities like training and marketing
maintain
monitoring, evaluating, marketing, other ongoing housekeeping tasks
SPARKDAWN MEDIA

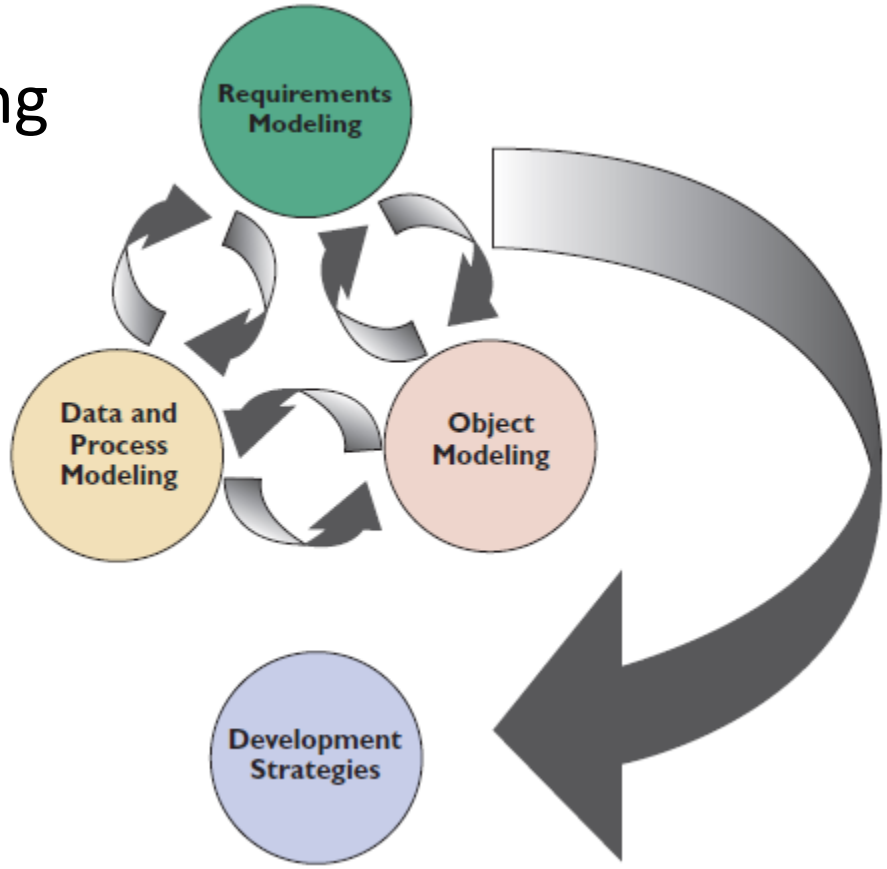
System Analysis Phase Overview

- The overall objective of the systems analysis phase is to understand the proposed project, ensure that it will support business requirements, and build a solid foundation for system development.
- In this phase, we use models and other documentation tools to visualize and describe the proposed system.

Systems Analysis Activities

- The systems analysis phase includes the four main activities :

1. **Requirements Modeling**
2. Data and Process Modeling
3. Object modeling
4. Development strategies.



The System Requirements

- **In Requirements modeling,** which involves **fact-finding** to describe the current system and identification of the requirements for the new system, such as:

1. Inputs
2. Outputs
3. Processes
4. Performance
5. Control/Security

System Requirements Checklist

1. Inputs

- Manufacturing employees must swipe their ID cards into online data collection terminals that record labor costs and calculate production efficiency
- The department head must enter overtime hours on a separate screen

System Requirements Checklist

2. Outputs

- The Web site must report online volume statistics every four hours, and hourly during peak periods
- The inventory system must produce a daily report showing the part number, description, quantity on hand, quantity allocated, quantity available, and unit cost of all sorted by part number

System Requirements Checklist

3. Processes

- The student records system must calculate the GPA at the end of each semester
- As the final step in year-end processing, the payroll system must update employee salaries, bonuses, and benefits and produce tax data required by the IRS

System Requirements Checklist

4. Performance

- The system must support 25 users online simultaneously
- Response time must not exceed four seconds

System Requirements Checklist

5. Controls

- The system must provide log-on security at the operating system level and at the application level
- An employee record must be added, changed, or deleted only by a member of the human resources department

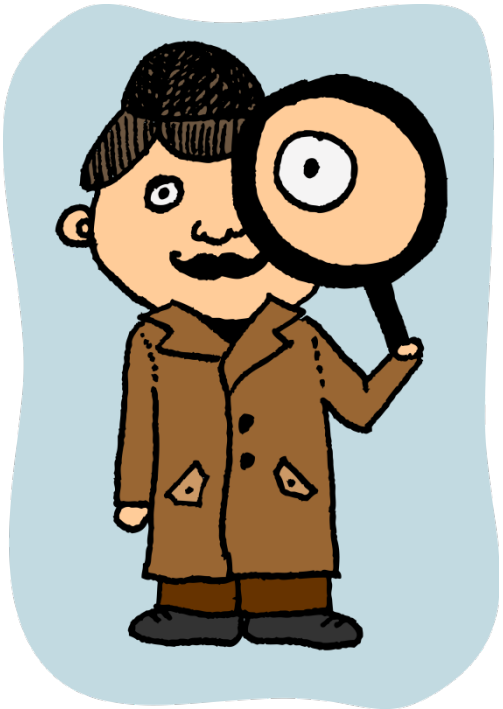
Fact-Finding

- Fact-Finding Overview
  - Begin to **collect information**
  - The first step is to identify the information, you need to develop a fact-finding plan.
- Who, What, Where, When, How, and Why?

- Fact-finding technique involves answers to five familiar questions: *who*, *what*, *where, when*, and *how*.
- For each of those questions, you also must ask another very important question which is ***"why".***

Fact-Finding

- The techniques used in fact finding:
  1. Interviews
  2. Documentation review
  3. Observations
  4. Questionnaires
  5. Sampling
  6. Research



1. Interviews

- An **interview** is a planned meeting during which obtain information from another person.
- We must have the skills needed to plan, conduct, document, and evaluate interviews successfully.
- After identify the information needed, we can begin the interviewing process, which consists of seven steps for each.

Interview

- **Step 1: Determine the People to Interview**
  - Informal structures
- **Step 2: Establish Objectives for the Interview**
  - Determine the general areas to be discussed
  - List the facts you want to gather

Interviews

- **Step 3: Develop Interview Questions**
  - Creating a standard list of interview questions helps to keep you on track and avoid unnecessary tangents
  - Avoid leading questions
  - Open-ended questions
  - Closed-ended questions
  - Range-of-response questions

Interviews

- **Step 4: Prepare for the Interview**
  - Careful preparation is essential because interview is an important meeting and not just a casual chat
  - Limit the interview to no more than one hour
  - Send a list of topics
  - Ask the interviewee to have samples available

Interviews

- **Step 5: Conduct the Interview**
  - Develop a specific plan for the meeting
  - Begin by introducing yourself, describing the project, and explaining interview objectives
  - Use engaged listening
  - Allow the person enough time to think about the question
  - After interview, summarize the session and seek a confirmation

Interviews

- **Step 6: Document the Interview**
  - Note taking should be kept to a minimum
  - After the interview, record the information quickly
  - After the interview, send memo expressing appreciation, including the main points discussed so the interviewee has a written summary and can offer additions or corrections

Interviews

- **Step 7: Evaluate the Interview**
  - In addition to recording the facts obtained in an interview, try to identify any possible biases

2. Document Review

– Help you understand how the current system is supposed to work

– Remember that system documentation sometimes is out of date.

- Forms can change or be discontinued, and documented procedures often are modified or eliminated.
  - Example: if the system uses a software package, you should review the documentation for that software.

3. Obrservation

- Seeing the system in action gives you additional perspective and a better understanding of the system procedures.
- Personal observation also allows you to verify statements made in interviews and determine whether procedures really operate as they are described.
- Through observation, you might discover that neither the system documentation nor the interview statements are accurate.

Obrservation

- Observation can provide knowledge needed to test or install future changes and can help build relationships with the users who will work with the new system.
- Plan your observations in advance by preparing a checklist of specific tasks you want to observe and questions you want to ask.

4. Questionnaires

– Also called a survey is a document containing a number of standard questions that can be sent to many individuals.

– When designing a questionnaire, the most important rule of all is to make sure that your questions collect the right data in a form that you can use to further your fact-finding.

– Used to obtain information about wide range of topics, difficulties and opinions of how the job could be performed better or more efficiently.

Sampling

- Sampling techniques include:-
  - Systematic sample
  - Stratified sample
  - Random sample
  - Main objective of a sample is to ensure that it represents the overall population accurately

Research

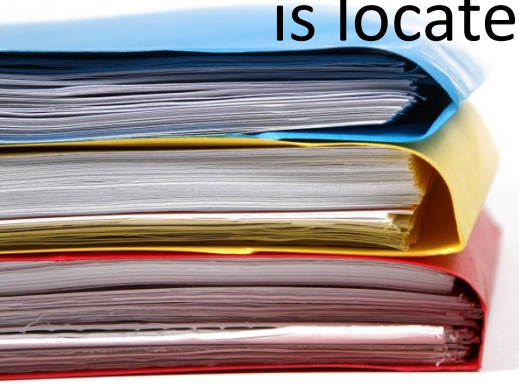
- Can include the Internet, IT magazines, and books to obtain background information, technical material, and news about industry trends and developments

Interviews versus Questionnaires

- Interview is more familiar and personal
- Questionnaire gives many people the opportunity to provide input and suggestions
- Brainstorming
- Structured brainstorming
- Unstructured brainstorming

Documentation

- The Need for Recording the Facts
  - Record information as soon as you obtain it
  - Use the simplest recording method
  - Record your findings in such a way that they can be understood by someone else
  - Organize your documentation so related material is located easily

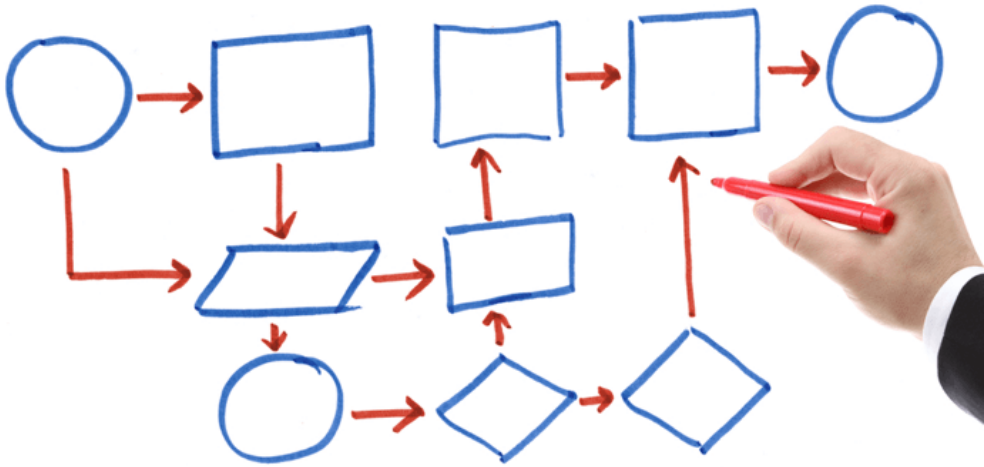


Documentation

- Software Tools
  - CASE Tools
  - Productivity Software
    - Word processing, spreadsheets, database management, presentation graphics programs
    - Histogram

Preview of Data and Process Modeling

- At the conclusion of requirements modeling, systems developers should have clear understanding of business processes and system requirements
- The next step is to model the logical design of the system



Functional Decomposition Diagram (FDD)

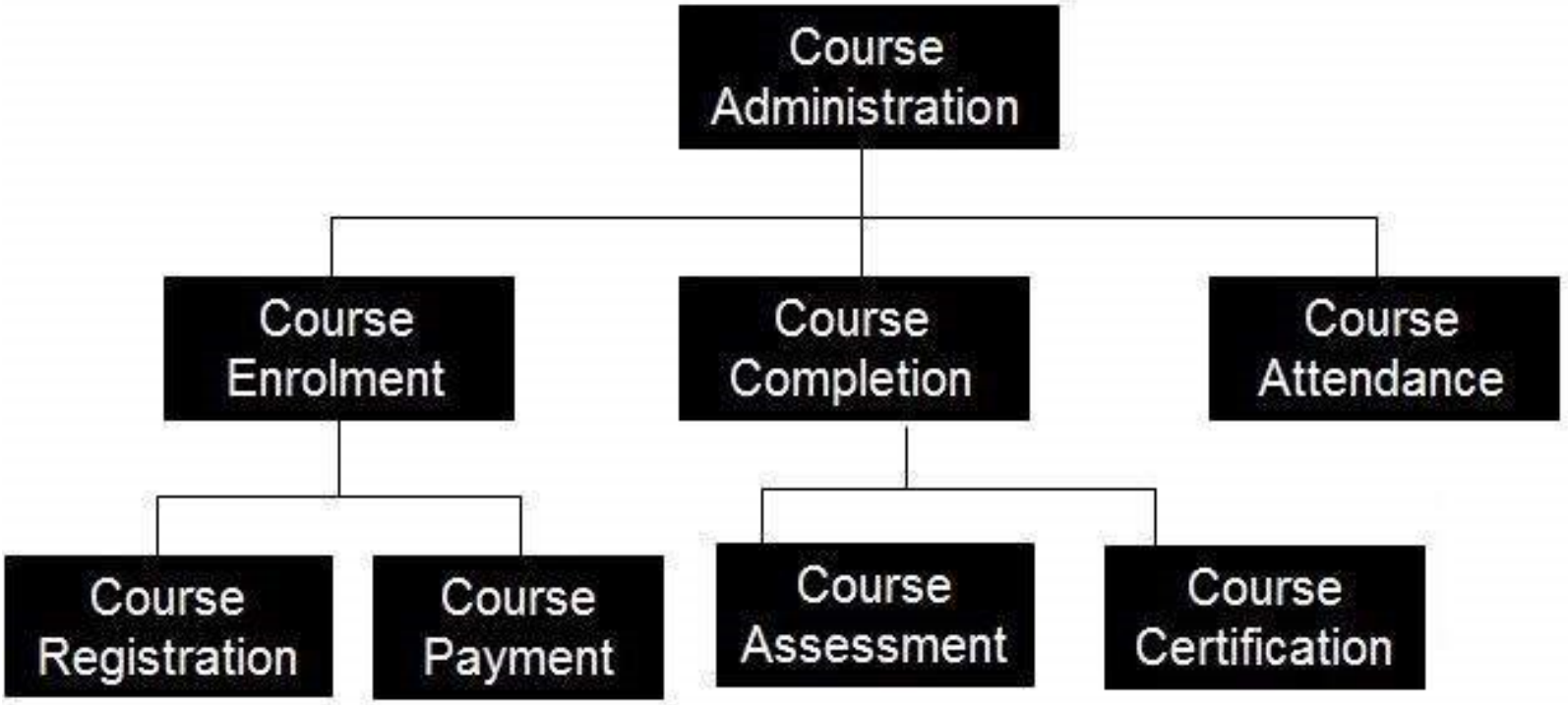
- Functional decomposition corresponds to the various functional relationships as how the original complex business function was developed.
- It mainly focusses on how the overall functionality is developed and its interaction between various components.
- Large or complex functionalities are more easily understood when broken down into pieces using functional decomposition.

When & How?

- Functional Decomposition is mostly used during the project analysis phase in order to produce functional decomposition diagrams as part of the functional requirements document.
- Functional Decomposition is done after meeting with business analysts and subject matter expertise.
- Decompose the first level components with their functions and continue to decompose to lower levels until sufficient level of detail is achieved
- Perform an end-to-end walk-through of the business operation and check each function to confirm that it is correct.

FDD

Example, the best describes the Functional Decomposition:



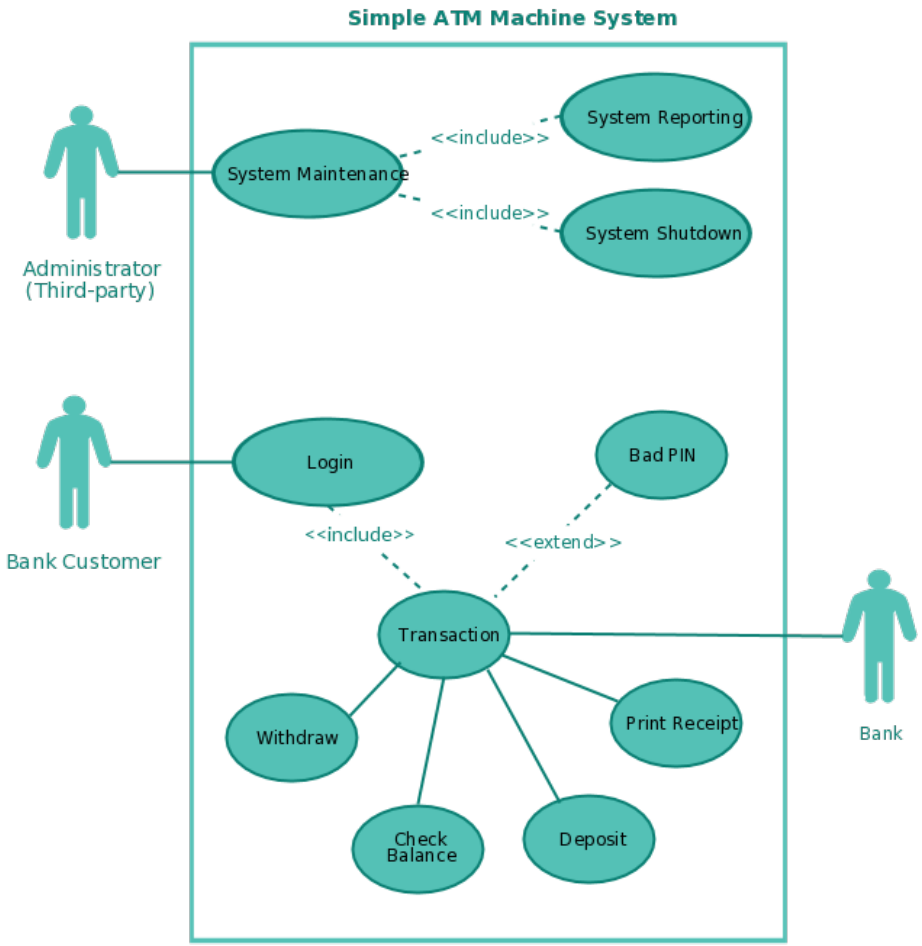
Unified Modelling Language (UML)

- **Unified Modeling Language (UML)** is a widely **used method** of visualizing and documenting software systems design.
- UML uses object-oriented design concepts, but it is independent of any specific programming language and can be used to describe business processes and requirements generally.

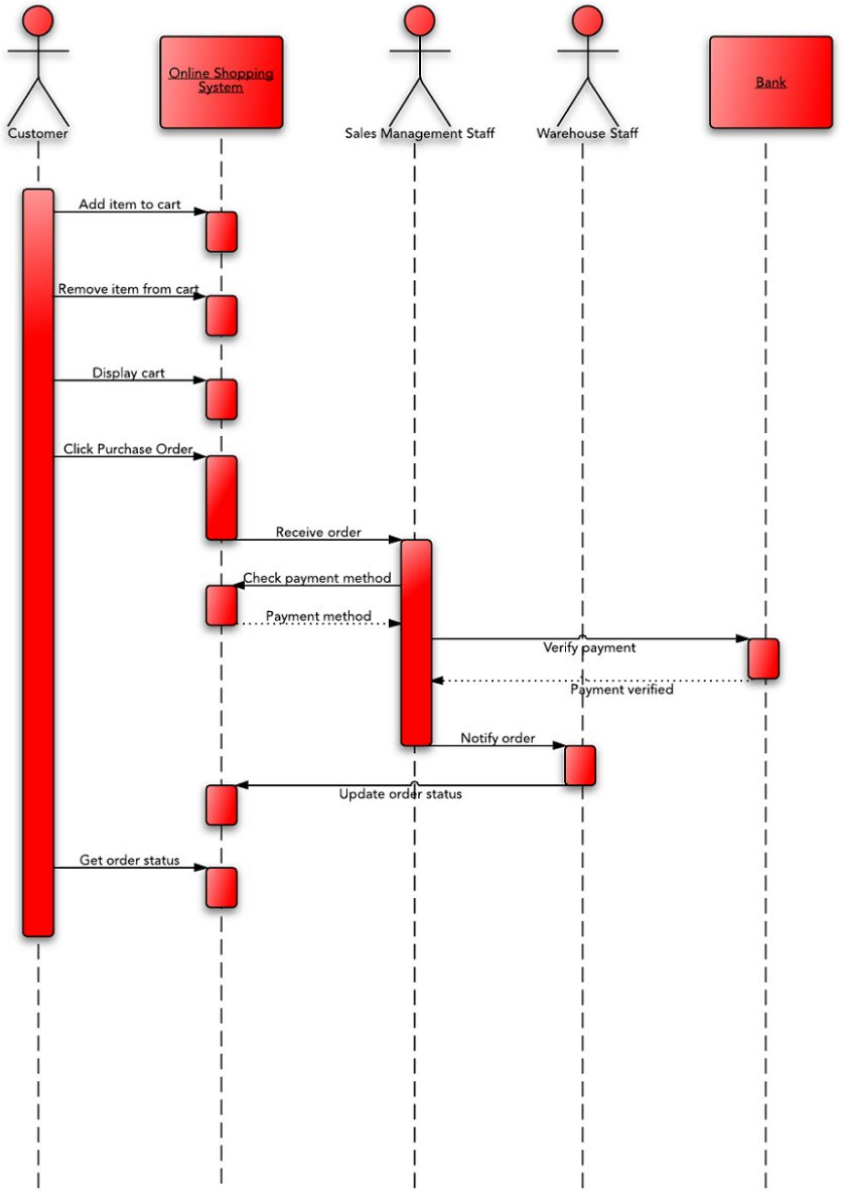
Unified Modelling Language (UML)

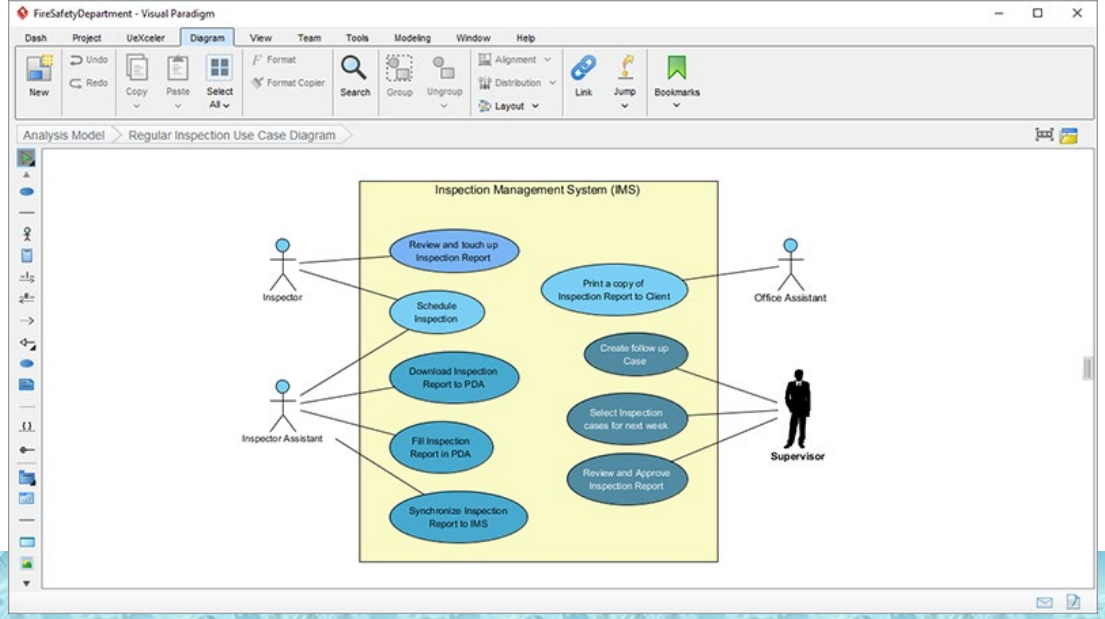
- UML provides various graphical tools, such as **use case diagrams and sequence diagrams**.
- During requirements modeling, a systems analyst can utilize the UML to represent the information system from a user's viewpoint.

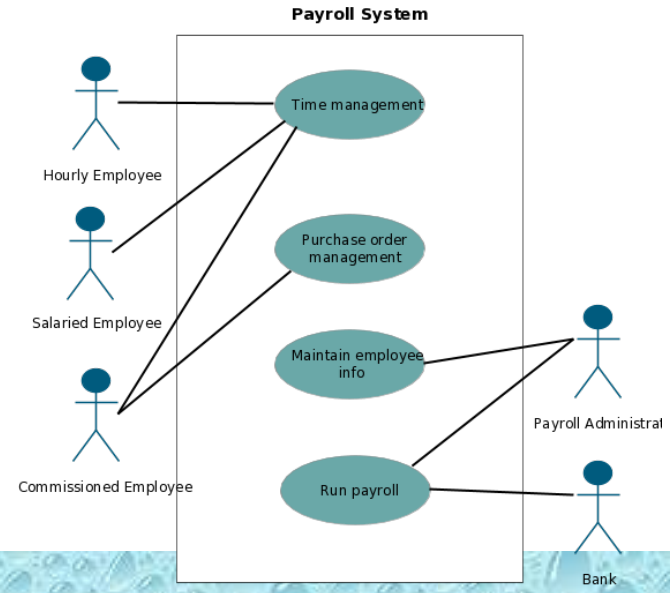
Use Case Diagram



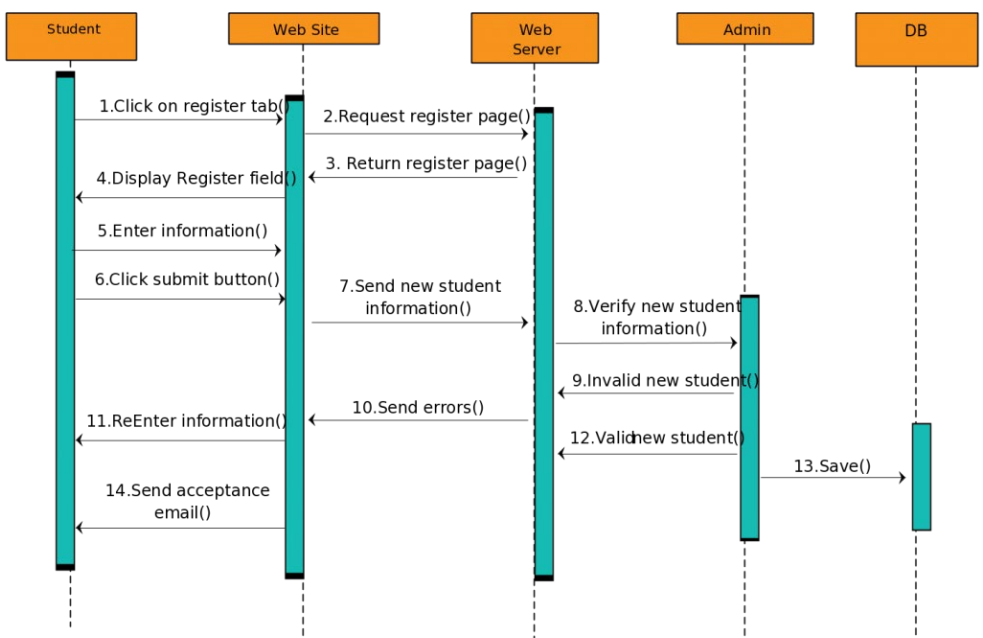
Sequence Diagram

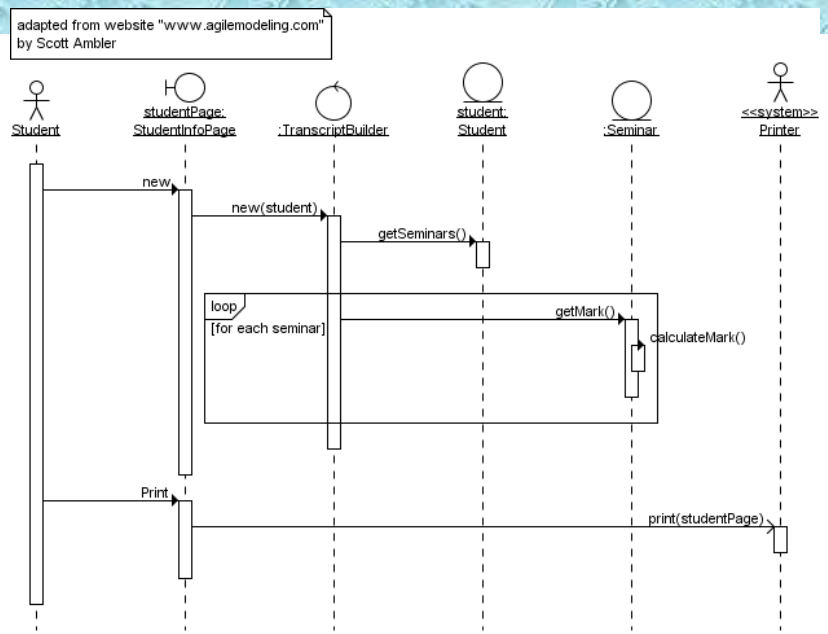






Let's do some activities





Learning outcomes

3.2 Understand Data and Process Modelling.

- Describe data and process modelling concepts and tools.
- Symbols used in data flow diagrams
- Draw data flow diagrams
- Explain how to level and balance a set of data flow diagrams.
- Data dictionary
- Relationship logical and physical models.
- Draw Entity Relationship Diagram.

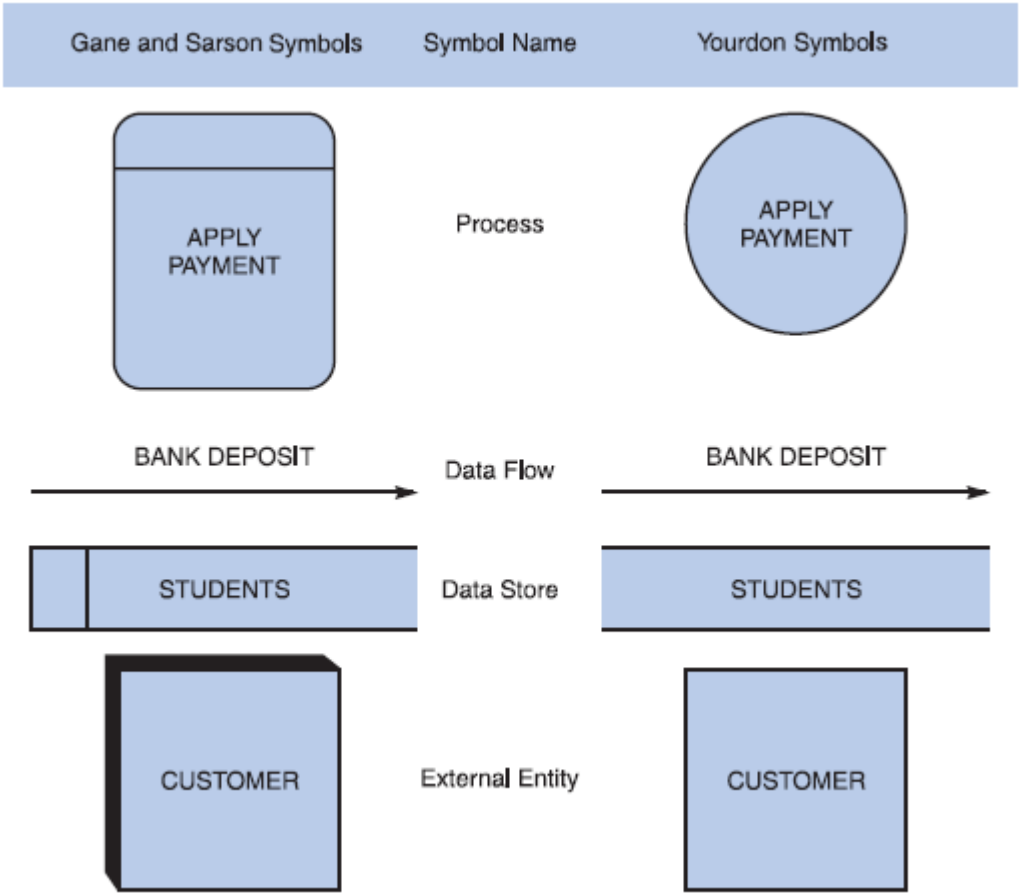
Overview of Data and process Modelling Tools.

- System analysts use many graphical technique to describe an information system.
- A data flow diagram (DFD) **uses various symbols** to show how the system transforms input data into useful information.
- A data flow diagram (DFD) shows how data moves through an information system but does not show program logic or processing steps.

Data Flow Diagrams

- A set of DFDs provides a logical model that shows what the system does, not how it does it.
- The **Data Dictionary** is an overall storehouse of information about the system, and serves as a central clearinghouse for all documentation
- **Process Descriptions** to explain the logical steps that each process performs. To create these descriptions, we use three tools: structured English statements, decision tables, and decision trees.

Data Flow Diagrams



- **DFD Symbols**

DFD Symbols

- **Process Symbol**

- Receives input data and produces output that has a different content from or both.

- Contain the business logic also called business rules.

- Referred to as a black box.

DFD Symbols

- **Data Flow Symbol**

- Represents one or more data items

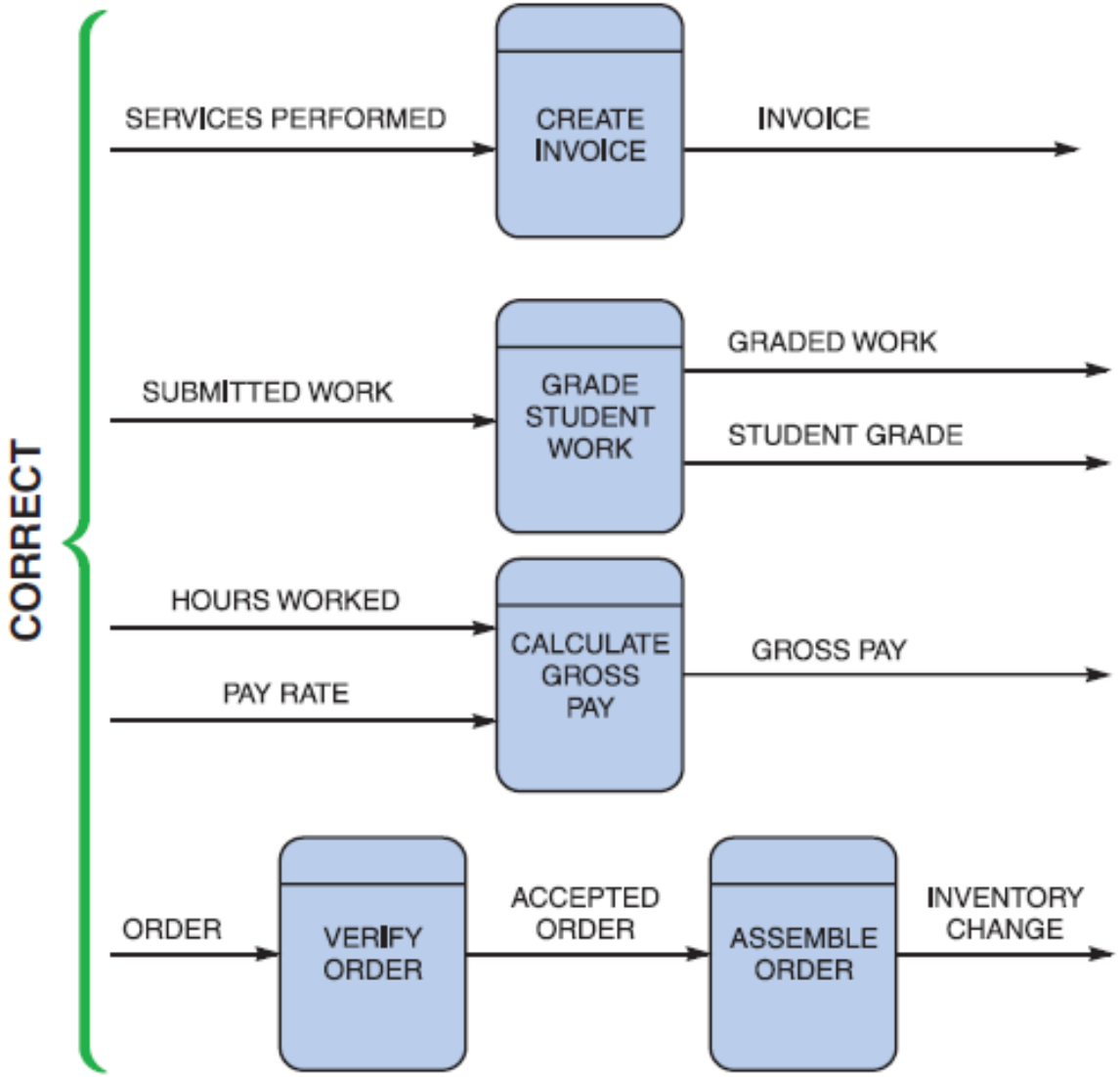
- The symbol for a data flow is a line with a single or double arrowhead

- Spontaneous generation

- Black hole

- Gray hole

Examples of correct combinations of data flow and process symbols.



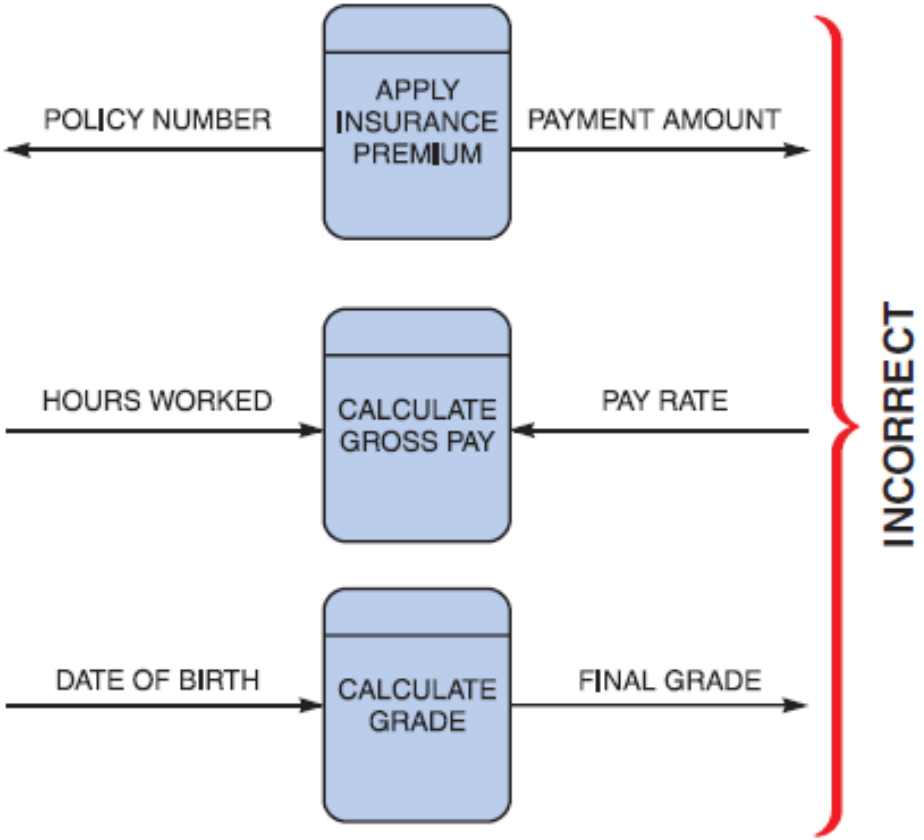
Examples of incorrect combinations of data flow and process symbols

APPLY INSURANCE PREMIUM has no input and is called a spontaneous generation process.

CALCULATE GROSS PAY has no outputs and is called a black hole process.

CALCULATE GRADE has an input that is obviously unable to produce the output.

This process is called a gray hole.



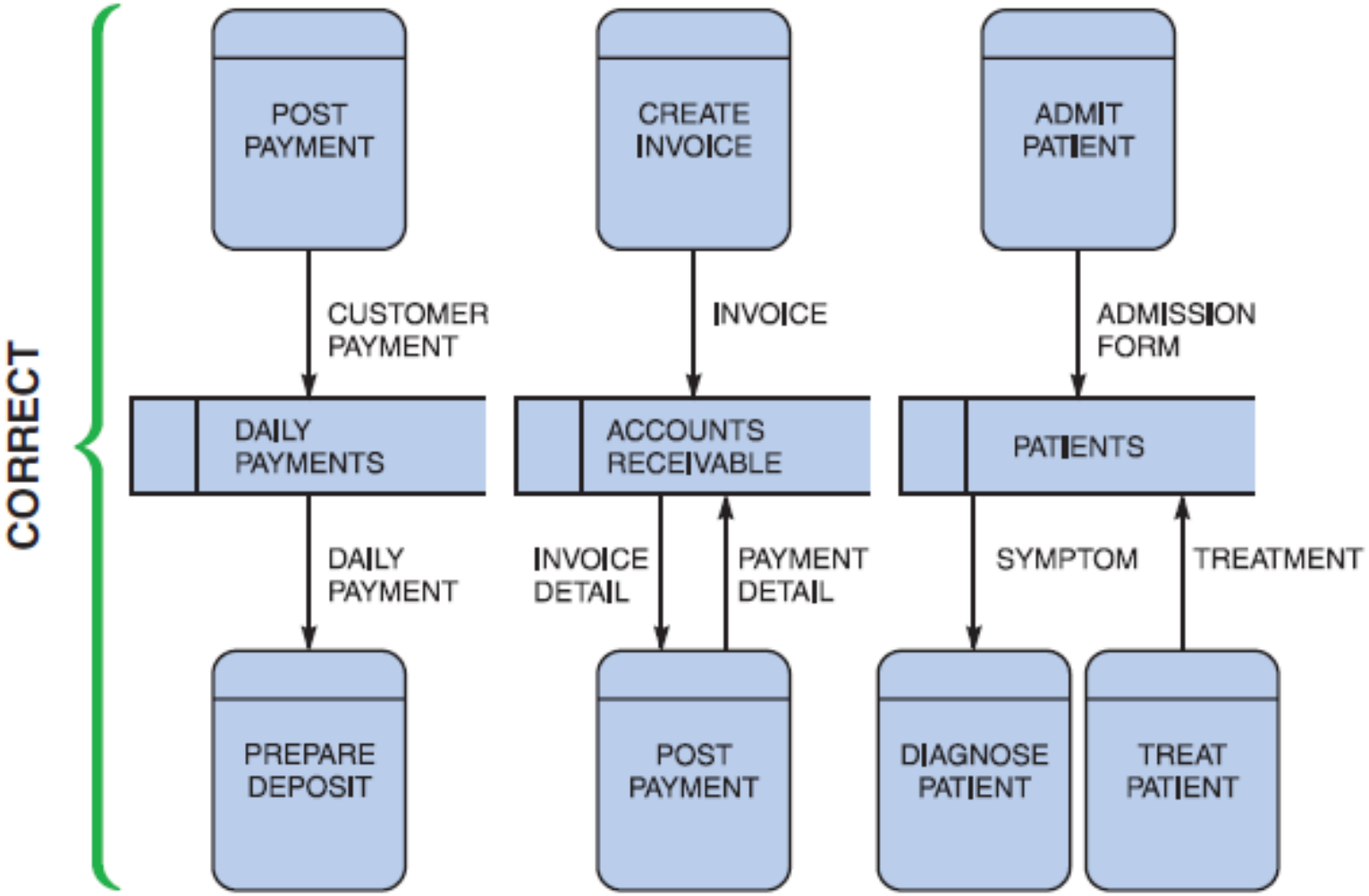
DFD Symbols

- **Data Store Symbol**

- Represent data that the system stores.

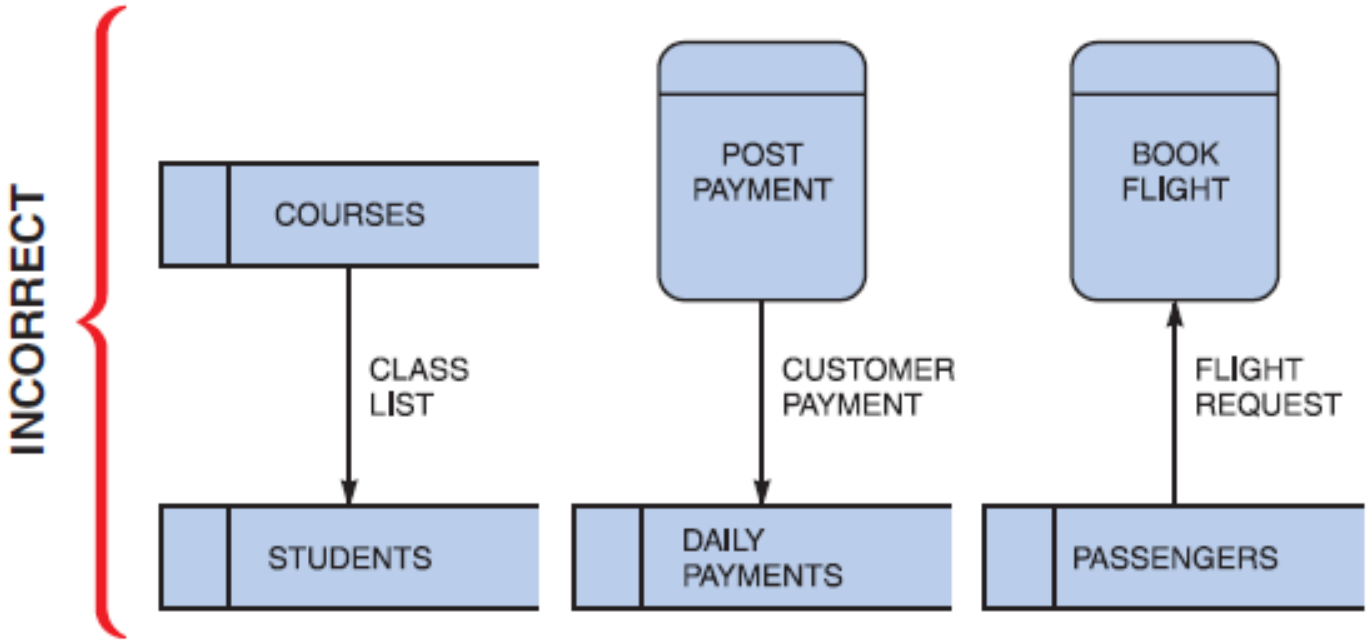
- The physical characteristics of a data store are unimportant because you are concerned only with a logical model.

Examples of correct uses of data store symbols in a data flow diagram



Examples of incorrect uses of data store symbols

- Two data stores cannot be connected by a data flow without an intervening process, and each data store should have an outgoing and incoming data flow.



DFD Symbols

- **Entity Symbol**

- Name of the entity appears inside the symbol.

- A DFD shows only external entities that provide data to the system or receive output from the system.

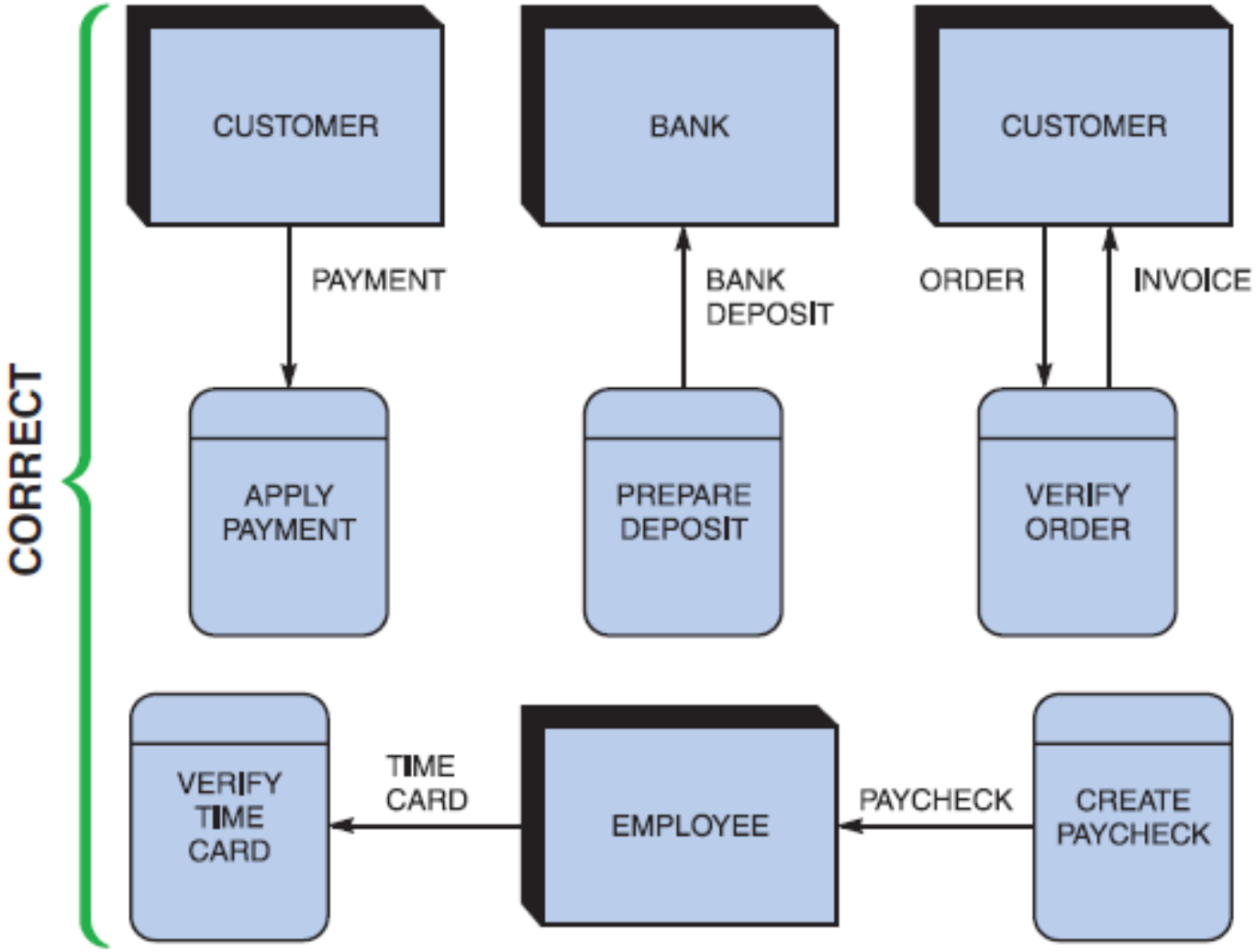
- A DFD shows the boundaries of the system and how the system interfaces with the outside world.

- DFD entities also are called **terminators**, because they are data origins or final destinations.

Entity Symbol

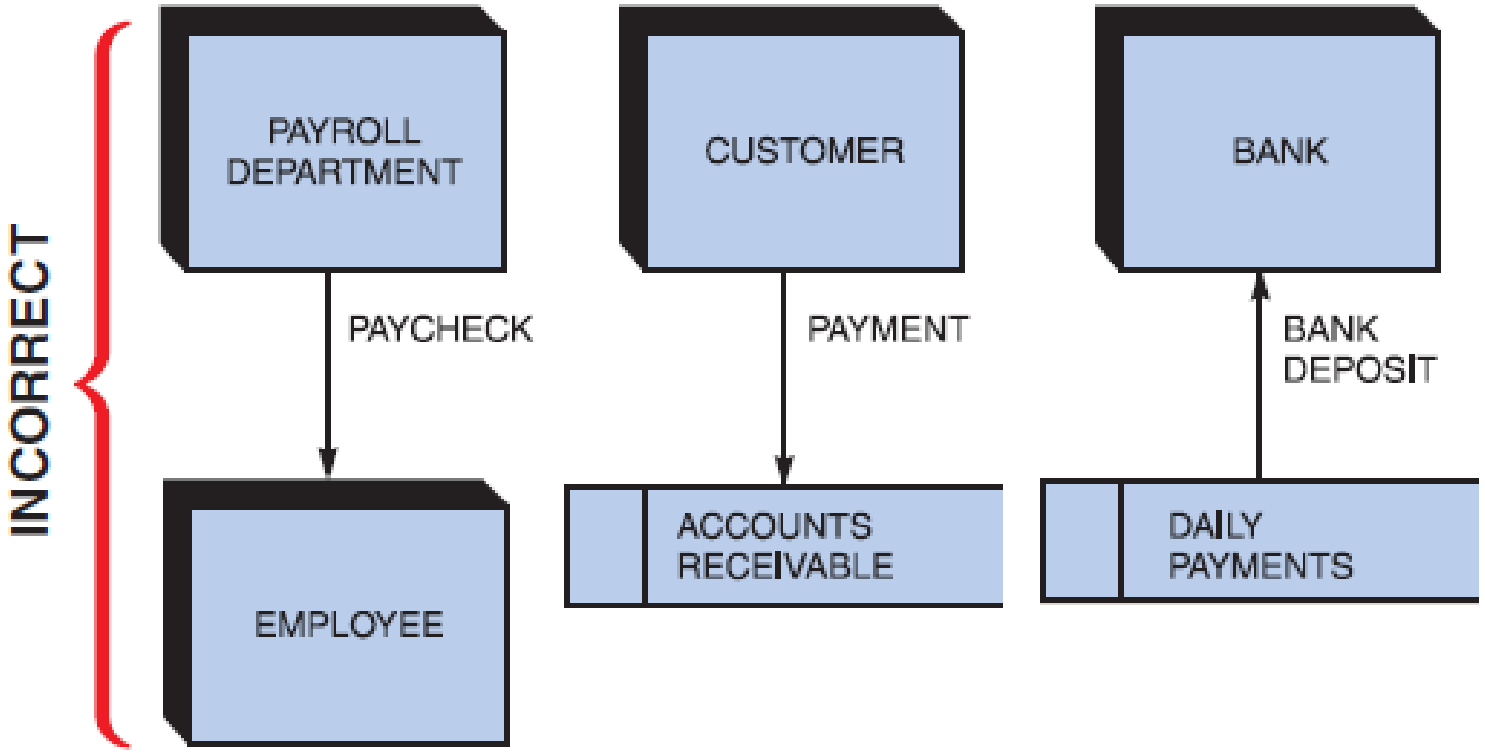
- Systems analysts call an entity that supplies data to the system a **source**, and an entity that receives data from the system a **sink.**
- An external entity can be a source or a sink or both, but each entity must be connected to a process by a data flow.

Examples of correct uses of external entities in a data flow diagram

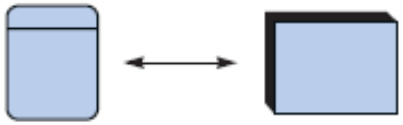
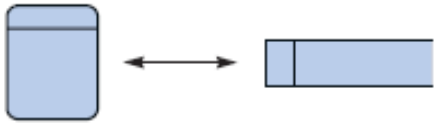
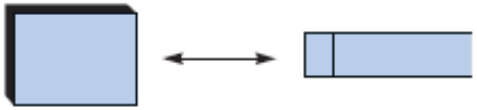


Examples of incorrect uses of external entities

- An external entity must be connected by a data flow to a process, and not directly to a data store or to another external entity.



E uses of data

Correct and Incorrect Examples of Data Flows		
	Process to Process	✓
	Process to External Entity	✓
	Process to Data Store	✓
	External Entity to External Entity	✗
	External Entity to Data Store	✗
	Data Store to Data Store	✗

Creating a Set of DFDs

- Create a graphical model of the information system based on your fact-finding results
- First, you will review a set of guidelines for drawing DFDs. Then you will learn how to apply these guidelines and create a set of DFDs using a three-step process

Creating a Set of DFDs

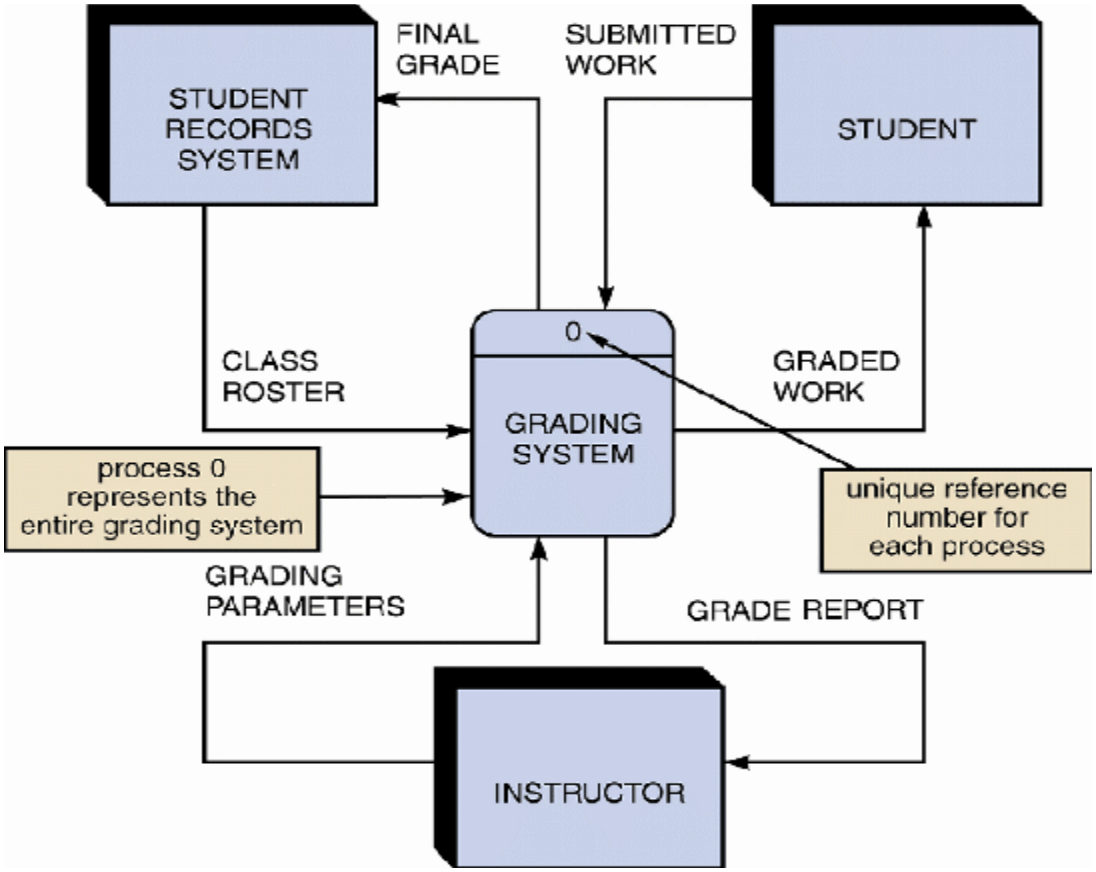
- Guidelines for Drawing DFDs
  - Draw the context diagram so that it fits on one page
  - Use the name of the information system as the process name in the context diagram
  - Use unique names within each set of symbols

Creating a Set of DFDs

- Guidelines for Drawing DFDs
  - Do not cross lines
  - Provide a unique name and reference number for each process
  - Obtain as much user input and feedback as possible

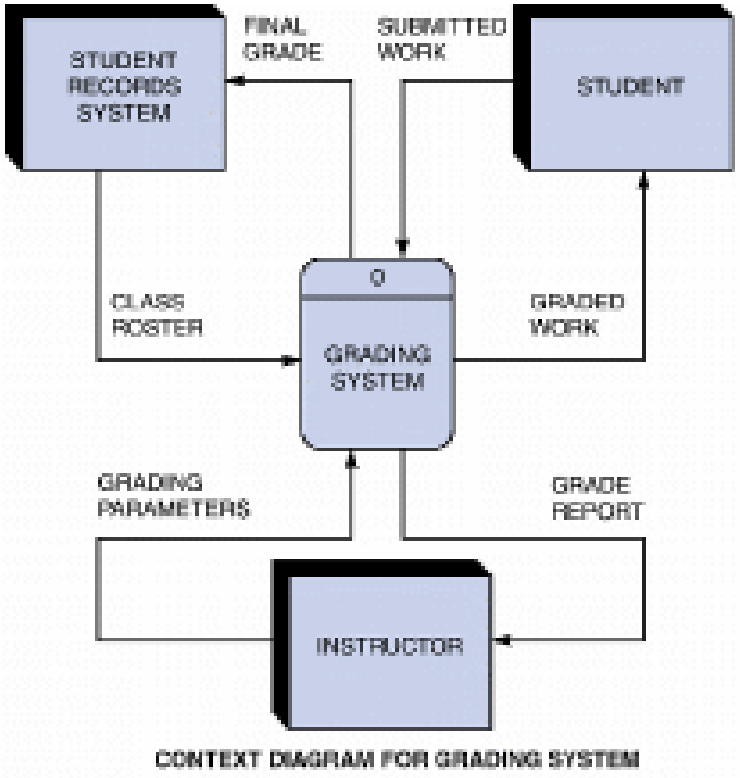
Creating a Set of DFDs

- Step 1: Draw a Context Diagram



Creating a Set of DFDs

- Step 2: Draw a Diagram 0 DFD



CONTEXT DIAGRAM FOR GRADING SYSTEM

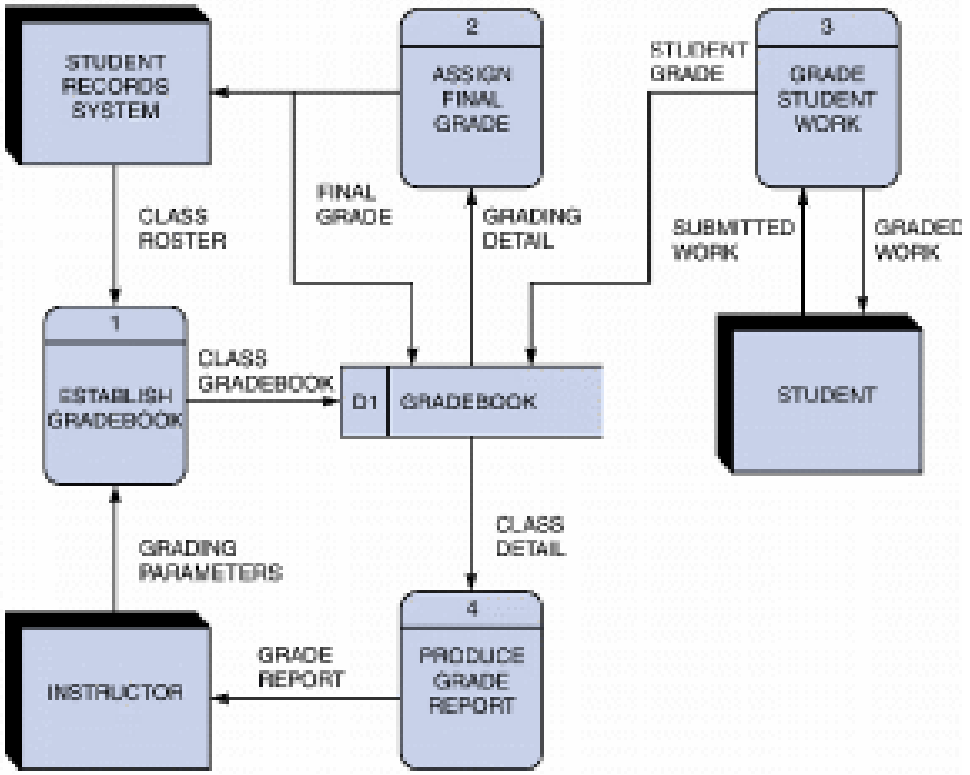


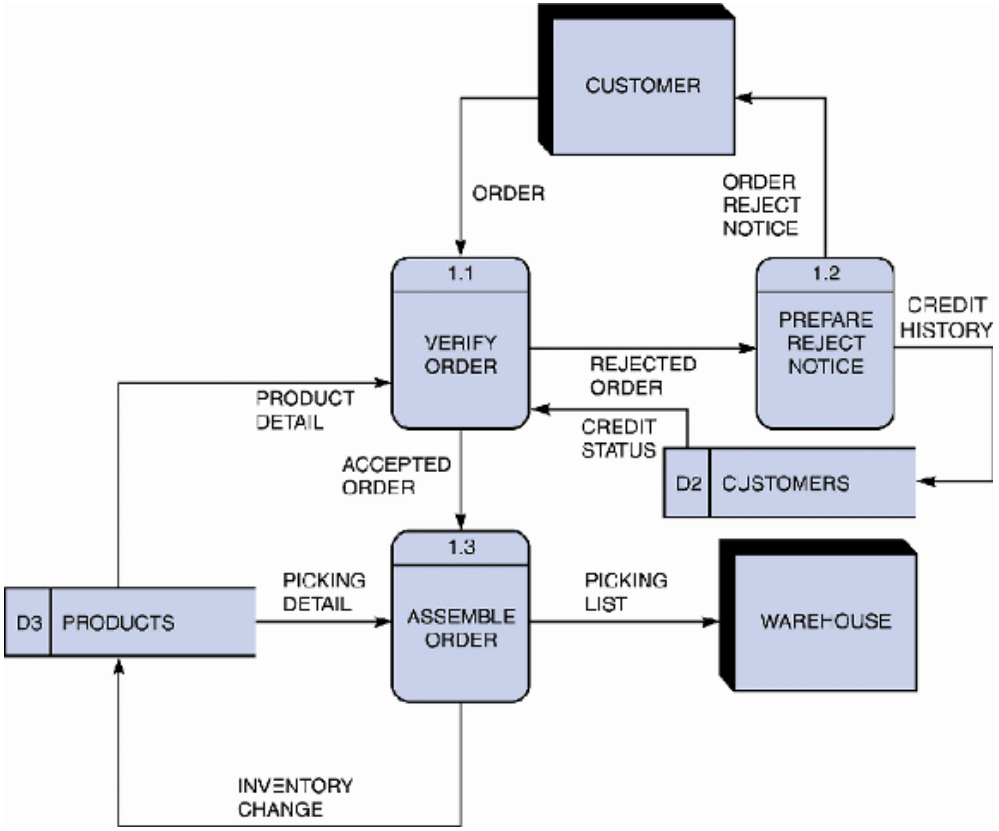
DIAGRAM 0 FOR GRADING SYSTEM

Creating a Set of DFDs

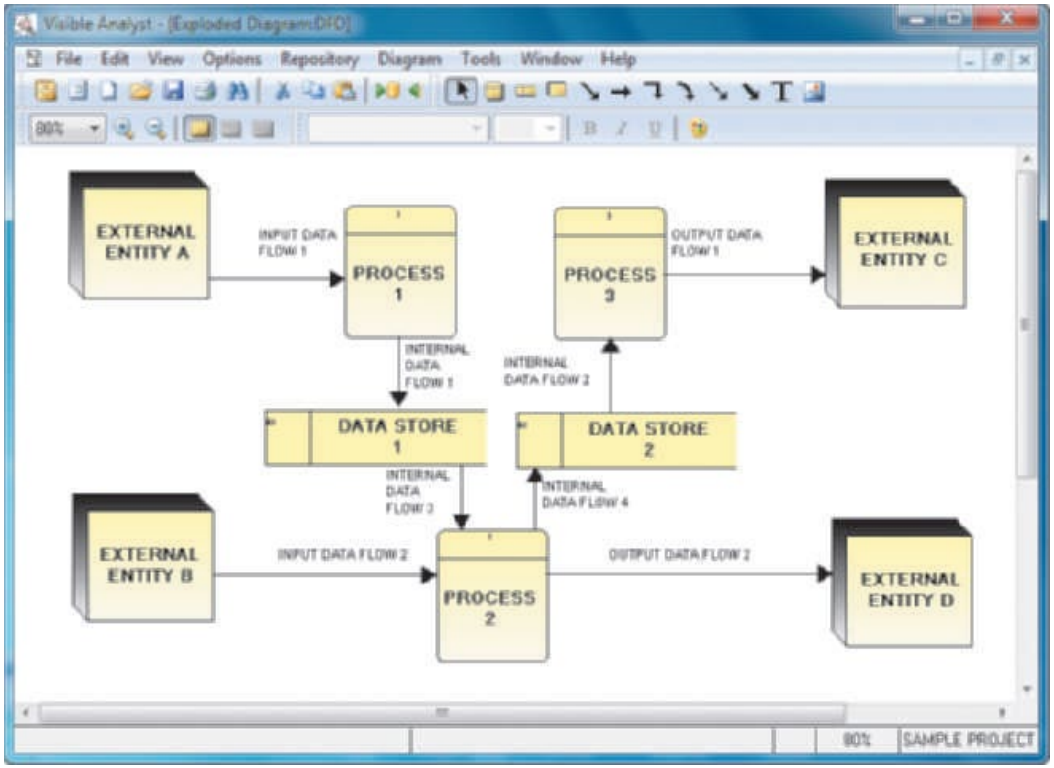
- Step 2: Draw a Diagram 0 DFD
  - If same data flows in both directions, you can use a double-headed arrow
  - Diagram 0 is an exploded view of process 0
  - Parent diagram
  - Child diagram
  - Functional primitive

Creating a Set of DFDs

- Step 3: Draw the Lower-Level Diagrams
  - Must use leveling and balancing techniques
  - Leveling examples
    - Uses a series of increasingly detailed DFDs to describe an information system
    - Exploding, partitioning, or decomposing



Creating a Set of DFDs



- Step 3: Draw the Lower-Level Diagrams
  - Balancing Examples
    - Ensures that the input and output data flows of the parent DFD are maintained on the child DFD

Data Dictionary

- A **data dictionary, or data repository**, is a central storehouse of information about the system's data.
- An analyst uses the data dictionary to collect, document, and organize specific facts about the system
- Also defines and describes all data elements and meaningful combinations of data elements

Data Dictionary

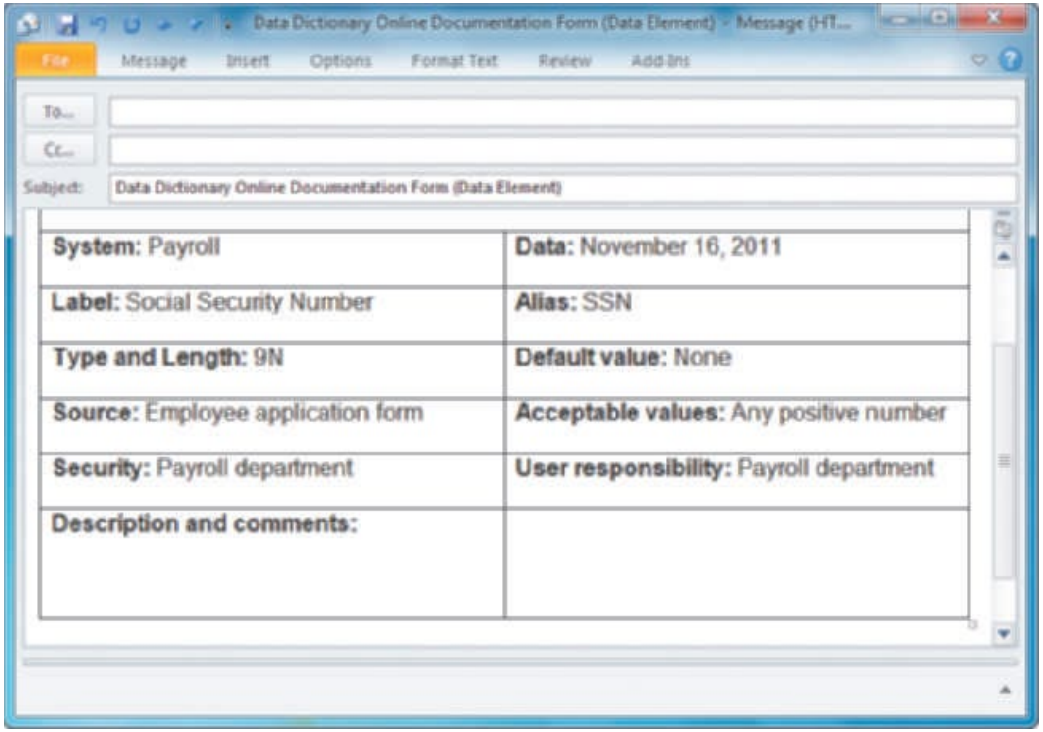
- A data element, also called a data item or field, is the smallest piece of data that has meaning.
- Data elements are combined into records, also called data structures.
- A record is a meaningful combination of related data elements that is included in a data flow or retained in a data store.

Data Dictionary

- Using CASE Tools for Documentation
  - The more complex the system, the more difficult it is to maintain full and accurate documentation
  - Modern CASE tools simplify the task
  - A CASE repository ensures data consistency
  - You will learn more about CASE tools in Part 2 of the Systems Analyst's Toolkit

Data Dictionary

- Documenting the Data Elements
  - You must document every data element in the data dictionary
  - The objective is the same: to provide clear, comprehensive information about the data and processes that make up the system



Data Dictionary

- Documenting the Data Elements
  - The following attributes usually are recorded and described
    - Data element name and label
    - Alias
    - Type and length
    - Default value
    - Acceptable values - Domain and validity rules

Data Dictionary

- Documenting the Data Elements
  - The following attributes usually are recorded and described
    - Source
    - Security
    - Responsible user(s)
    - Description and comments

Data Dictionary

- Documenting the Data Flows
  - The typical attributes are as follows
    - Data flow name or label
    - Description
    - Alternate name(s)
    - Origin
    - Destination
    - Record
    - Volume and frequency

Data Dictionary

- Documenting the Data Stores
  - Typical characteristics of a data store are
    - Data store name or label
    - Description
    - Alternate name(s)
    - Attributes
    - Volume and frequency

70

Data Dictionary

- Documenting the Processes
  - Typical characteristics of a process
    - Process name or label
    - Description
    - Process number
    - Process description

71

Data Dictionary

- Documenting the Entities
  - Typical characteristics of an entity include
    - Entity name
    - Description
    - Alternate name(s)
    - Input data flows
    - Output data flows

72

Data Dictionary

- Documenting the Records
  - Typical characteristics of a record include
    - Record or data structure name
    - Definition or description
    - Alternate name(s)
    - Attributes

73

Data Dictionary

- Data Dictionary Reports
  - Many valuable reports
    - An alphabetized list of all data elements by name
    - A report describing each data element and indicating the user or department that is responsible for data entry, updating, or deletion
    - A report of all data flows and data stores that use a particular data element
    - Detailed reports showing all characteristics of data elements, records, data flows, processes, or any other selected item stored in the data dictionary

74

Logical Versus Physical Models

- While structured analysis tools are used to develop a logical model for a new information system, such tools also can be used to develop physical models of an information system
- A physical model shows how the system's requirements are implemented

Logical Versus Physical Models

- Sequence of Models
  - Many systems analysts create a physical model of the current system and then develop a logical model of the current system before tackling a logical model of the new system
  - Performing that extra step allows them to understand the current system better

76

Logical Versus Physical Models

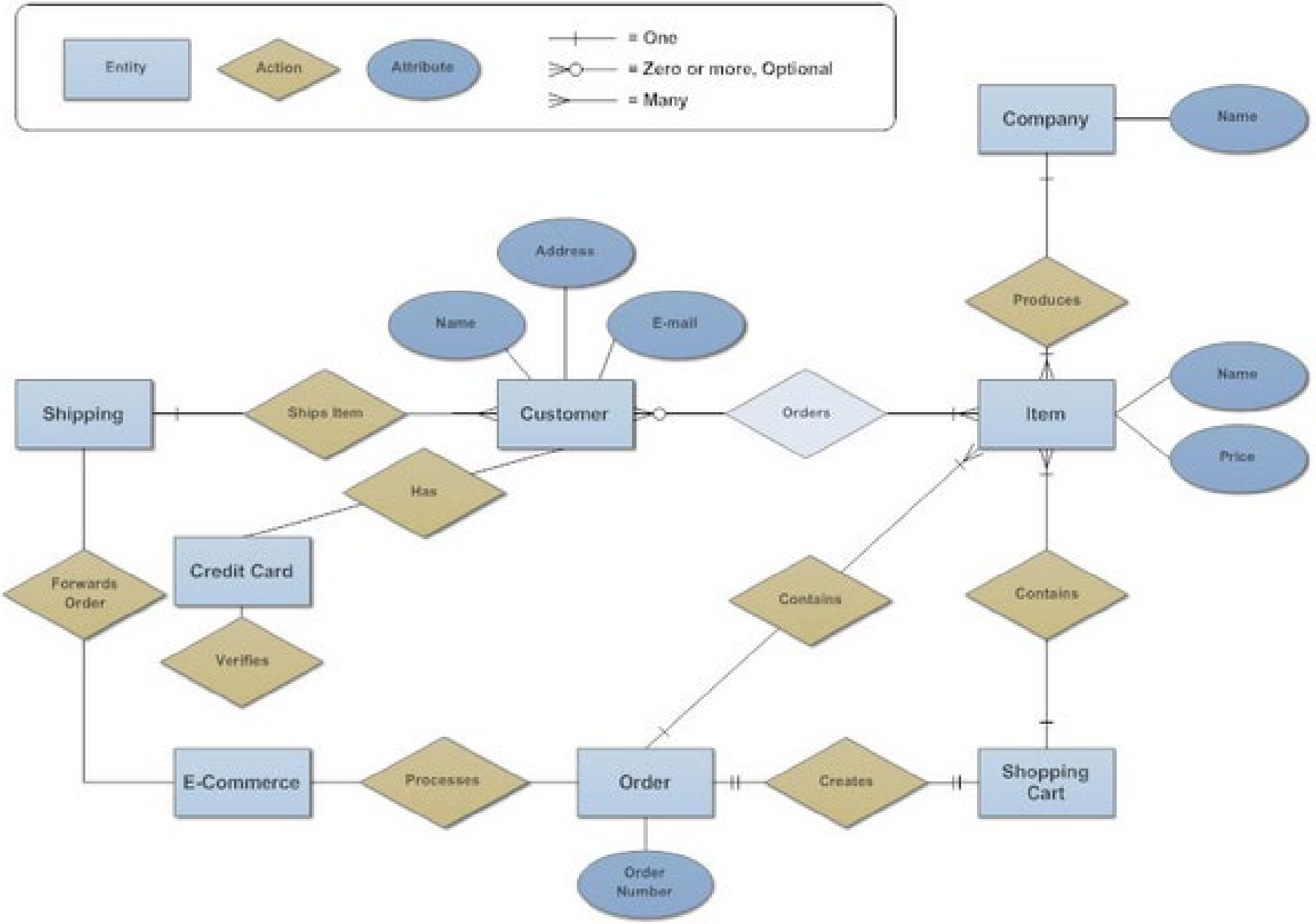
- Four-Model Approach
  - Develop a physical model of the current system, a logical model of the current system, a logical model of the new system, and a physical model of the new system
  - The only disadvantage of the four-model approach is the added time and cost

77

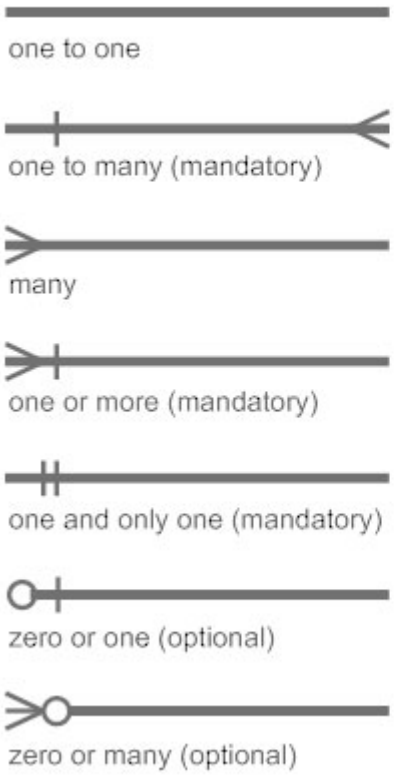
Entity Relationship Diagram (ERD)

- An **entity relationship diagram (ERD)** shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.
- At first glance an entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols, that make it unique.

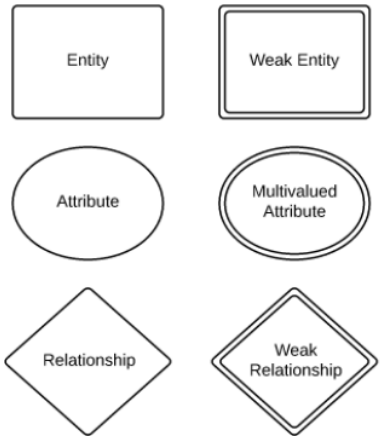
Entity Relationship Diagram - Internet Sales Model



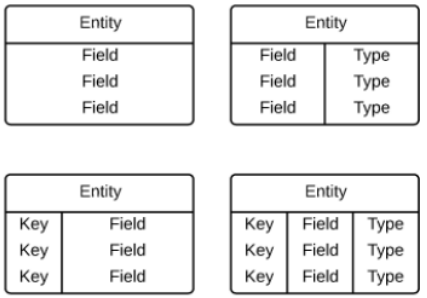
Information Engineering Style



Conceptual Data Model Symbols

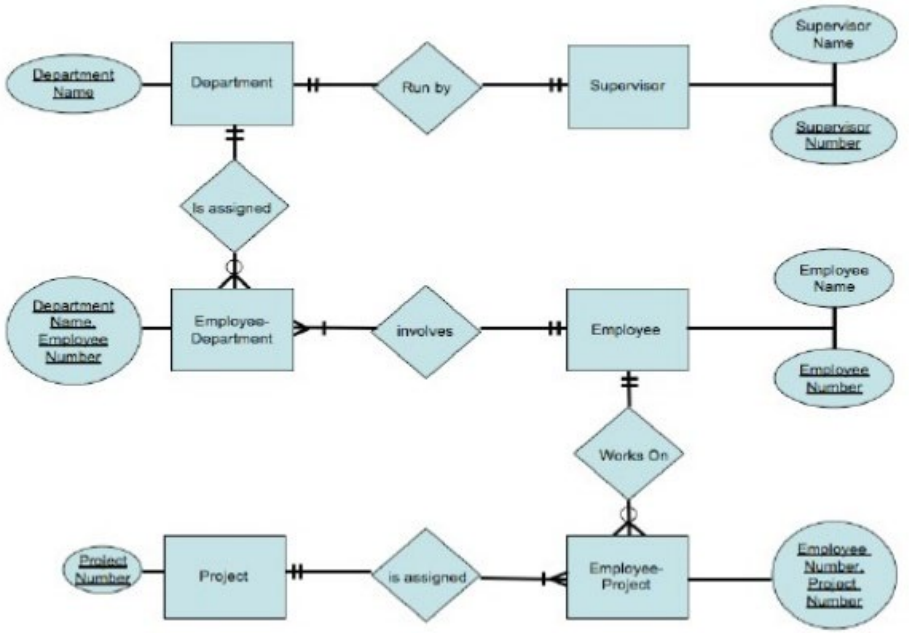


Physical Data Model Symbols



Let's do some activities

Draw Fully Attributed ERD



Good Luck