Data Science Joint Education Program

# INFO 200
# Information Systems Analysis

Chapter #2

# Understanding and Modelling Organizational Systems

# Keywords and Phrases (1)

| | |
|---|---|
| actor | feedback |
| associative entity | four levels of use cases |
| attributive entity | interdependent |
| closedness | interrelatedness |
| collaborative design | middle management |
| context-level data | openness |
| flow diagram | operations management |
| crow's foot notation | organizational boundaries |
| enterprise resource planning (ERP) | organizational culture |
| enterprise systems  entity (fundamental entity) | organizational subculture |
| entity-relationship (E-R) diagram | scope of the system |
| environment | Slack |

# Keywords and Phrases (2)

| | |
|---|---|
| strategic management | use case scenario |
| systems | virtual enterprise |
| use case | virtual organization |
| use case diagram | virtual team |

# Chapter #2 Overview

- An understanding that organizations and their members are systems and that analysts need to take a systems perspective
- Models organisational systems graphically using:
  - Context-level data flow diagrams
  - Entity-relationship diagrams (models)
  - Use cases
  - Use case scenarios
- Recognizes that different levels of management require different systems
- Provides an overview of organizational culture and an understanding that organizational culture influences the design of information systems
- Many technologies are available to support the creation of productive cultures and subcultures, including work-sanctioned social media sites such as Slack.

# Chapter #2 Overview

- To analyse and design appropriate information systems systems analysts must:
  - Understand the organizations they work in as systems are shaped through the interactions of three  main forces:
    - Levels of management
    - Design of organizations
    - Organizational cultures
- Organizations are large systems composed of interrelated subsystems:
  - The subsystems are influenced by three broad levels of management decisions:
    - Decision-makers
    - Operations
    - Middle management
    - Strategic management
  - The subsystems cut horizontally and vertically across the organizational system

# Organisations as Systems

# Organisations as Systems

- Organizations and their members are usefully conceptualized as systems designed to accomplish:
  - Predetermined goals and objectives
  - Using people and other resources that they (the organizations) employ
- Organizations are composed of systems and sub-systems which are:
  - Smaller interrelated systems (departments, units, divisions, and  so on) that serve specialized functions
  - Typical functions include:
    - Accounting / sales  / production / IT support / operations / legal / management.
    - Specialized functions (smaller systems) are  eventually reintegrated in various ways to form an effective organizational whole
  - Specialized functions are reintegrated to form an effective organizational whole

# Organisations as Systems

- The significance of conceptualizing organizations as complex systems is that systems principles allow insight into how organizations work.
- To ascertain information requirements properly (and to design appropriate information systems):
  - It is essential to understand the organization as a whole
- All systems are composed of subsystems (which include information systems)
  - Therefore when studying an organization we also examine how smaller systems are involved and  how they function

# Interrelatedness and Independence of Systems

- All systems and subsystems are interrelated and interdependent
  - This has important implications both for organizations and for systems analysts who seek to help them better achieve their goals
  - When any element of a system is changed or eliminated, the rest of the system’s elements and subsystems are also significantly affected – for example:
    - An organization may decide not to hire administrative assistants and to replace their functions with networked PCs
    - This decision has the potential to significantly affect not only the administrative assistants and the managers but also all the organization’s members who built up communications networks with the now-departed assistants

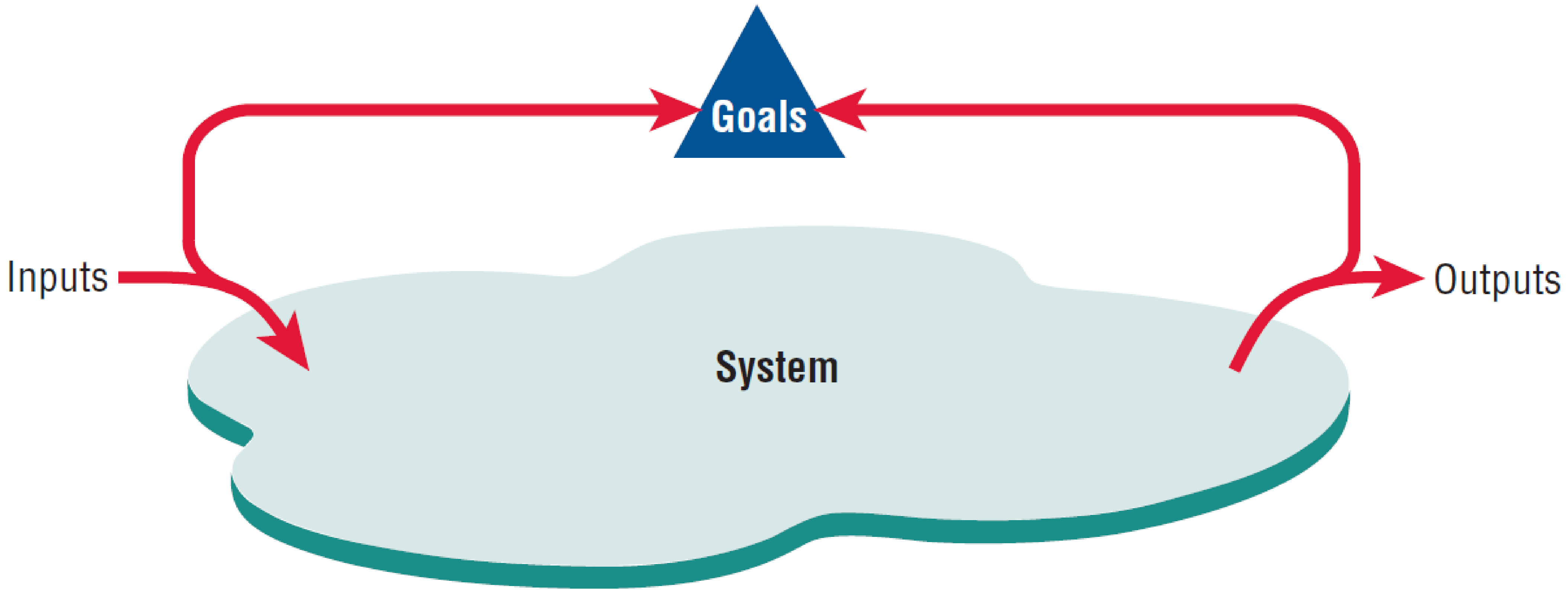
# Major Topics Introduced

- All systems process inputs from their environments
- By definition: processes change (or transform) inputs into outputs
- Whenever a system is examined:
    - Check and identify what is being changed or processed
    - If nothing is changed you may not be identifying a process
- Typical processes in systems include:
    - Verifying / updating / printing
- Another aspect of organizations as systems is that all systems are contained by boundaries separating them from their environments
    - Organizational boundaries exist on a continuum ranging from extremely permeable to almost impermeable.

# System Inputs and Outputs

- To continue to adapt and survive:
  - Organizations must be able first to import resources (*inputs*) through their boundaries
  - Organisations then use the resources to create products or services (*outputs*)
  - Organisations then exchange their finished products, services, or information with the outside world
- Feedback is one form of system control
- As systems, all organizations use planning and control to manage their resources effectively

# Figure 2.1



- Figure 2.1 shows how system outputs are used as feedback that compares performance with goals
- This comparison, in turn, helps managers formulate  more specific goals as inputs

# Feedback in Action

- Figure 2.1 shows how system outputs are used as feedback to compare actual performance with pre-defined goals
- The comparison is used to update and formulate new and possibly more specific goals as inputs
- For example:
  - A company that produces red white and blue weight-training sets along with gunmetal grey training sets
  - The company finds that one year after the summer Olympics very few red-white-and-blue sets are purchased.
  - Production managers use this information as feedback to make decisions about what quantities of each colour to produce
  - Feedback in this instance is useful for planning and control.

# Organisational Boundaries

- Ideal systems will be semi-autonomous and self-correct or self-regulate in such a way that decisions on typical occurrences are not required
- An example is a supply chain system for production planning that takes into account current and projected demand and formulates a proposed solution as output.
- An Italian knitwear manufacturer that markets its clothing in the United States has just such a system:
  - This company produces most of its sweaters in undyed white shades and uses its computerized inventory information system to find the best-selling colours
  - This information is used to dye sweaters immediately before shipping them
- Feedback is received from within the organization and from the outside environments around it.

# Major Topics Introduced

- Anything external to an organization’s boundaries is considered to be an environment
  - Numerous environments (with varying degrees of stability) constitute the global operating environment in which organizations exist and functions Including:
    1. The environment of the *community* in which the organization is physically located, which is shaped by the size of its population and its demographic profile, including factors such as education and average income
    2. The *economic* environment influenced by market factors including competition
    3. The *political* environment controlled through state and local governments
    4. The *legal* environment, issuing federal, state, regional, and local laws and guidelines.
- Although changes in environment status can be planned for, they often cannot be directly controlled by the organization
- The example cited in the book is for the USA.
  - Similar factors may be considered for other countries

# Open and Closed Organisations

- Related (and similar to) the concept of external boundary permeability is the concept of internal “openness” or “closedness” of organizations
- Open systems operate on:
  - Free flow of information
  - Output from one system becomes input to another system
- Closed systems operate on:
  - Restricted access to information
  - Limited by numerous rules
  - Information only on a “need to know” basis
- Openness and closedness also exist on a continuum
  - There is no such thing as an absolutely open or a completely closed organization

# Virtual Organizations and Virtual Teams

- Not all organizations or parts of organizations are visible in a physical location
- Entire organizations or units of organizations can now possess virtual components that enable:
  - Changes to configurations to adapt to changing project or marketplace demands
- Virtual enterprises:
  - Use networks of computers and communications technology to bring people with specific skills together electronically to work on projects that are not physically located in the same place

# Virtual Organizations and Teams

- Information  technology enables coordination of these remote team members.
- Virtual teams often spring up in  already-established organizations
- However in some instances
  - Organizations of remote workers have been able to succeed without the traditional investment in a physical facility
- The potential benefits of virtual organizations include:
  - Reducing costs of physical facilities
    - There are infrastructure (IT) costs
  - More rapid response to customer needs
    - Flexible remote working can increase customer service times

# Potential Social Benefits and Issues

- Virtual Organizations and Teams offer potential benefits to employees:
  - Remote working and flexible working (e.g., from home) may help virtual employees to fulfill their family obligations to growing children or aging parents
- There may however be cultural and organizational issues which make remote working impractical:
  - Collaboration and teamwork may become difficult (with flexible working)
  - Supervision and data security (with remote working) is a potential issue
  - Just how important it is to meet the social needs of virtual workers is still open to research and debate

# Distance Learning

- In considering virtual organizations:
  - An example may be distance learning where students were enrolled in a virtual university with no physical campus
- There are two important considerations:
  - Pedagogic systems
  - The cultural component
- An example of the cultural component:
  - Students requested items such as sweatshirts and coffee mugs etc with the virtual university's logo imprinted on them
  - These items are meaningful cultural artifacts that traditional brick-and-mortar schools have long  provided

# Collaborative Design

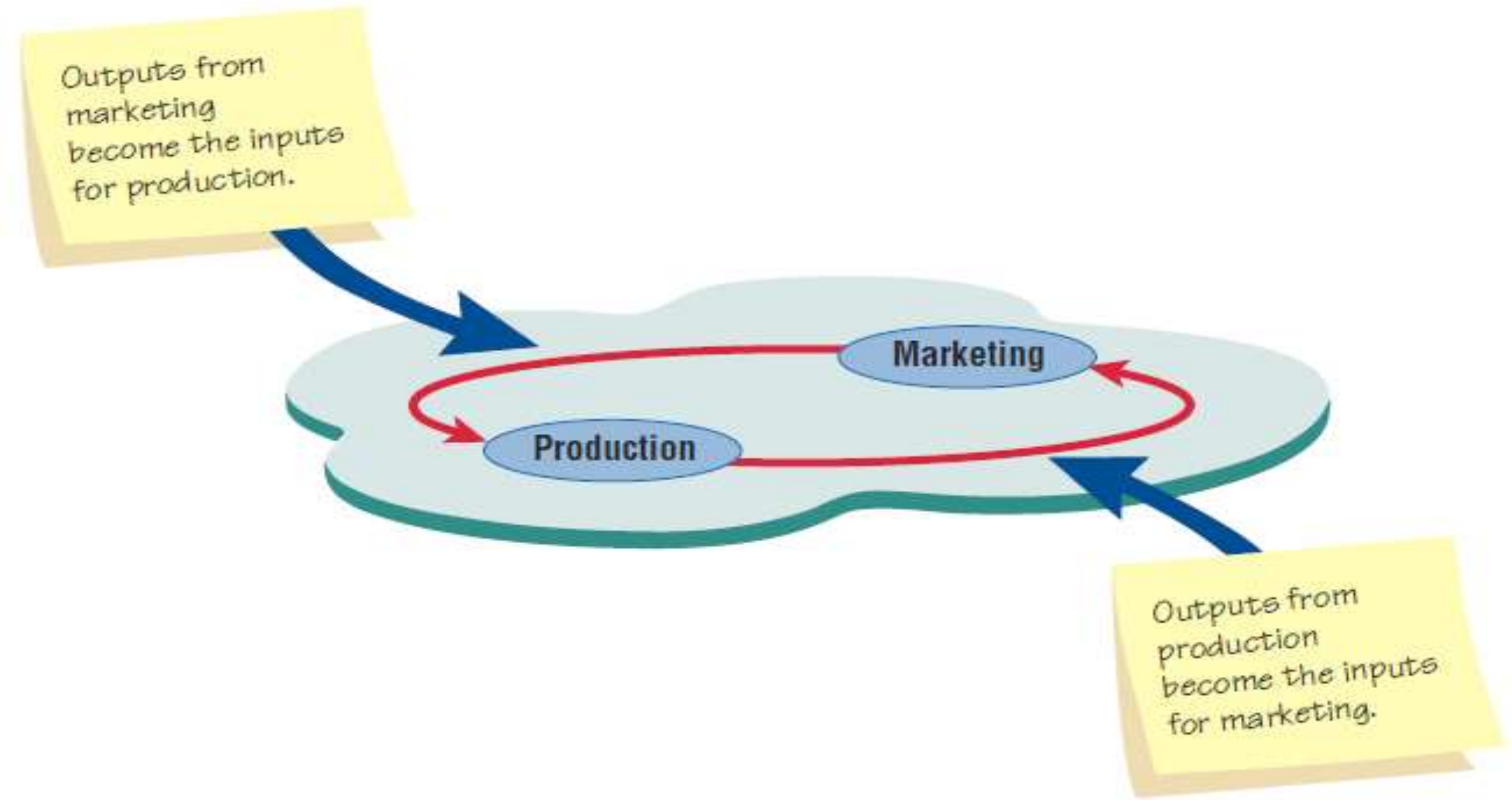
- Many systems analysis and design teams are now able to work virtually
  - Such collaboration may be national and increasingly international
  - Many such collaborations have provided an example for other types of employees to follow in accomplishing work virtually
- Some applications permit analysts who are providing technical assistance over the Web to “see”  the software and hardware configuration of the user requesting help
  - Thus creating an ad-hoc virtual team composed of the analyst and the user

# Taking a Systems Perspective

- Taking a systems perspective allows systems analysts to start broadly clarifying and understanding the various organizations
    - It is important that members of subsystems realize that their work is interrelated
    - Figure 2.2 shows:
        - How the outputs from the production subsystems serve as inputs for marketing
        - That the outputs of marketing serve as new inputs for production
        - Neither subsystem can properly accomplish its goals without the  other.
        - Problems occur when each manager possesses a different picture of the importance of their own functional subsystem
    - In Figure 2.3 models the marketing managers perspective

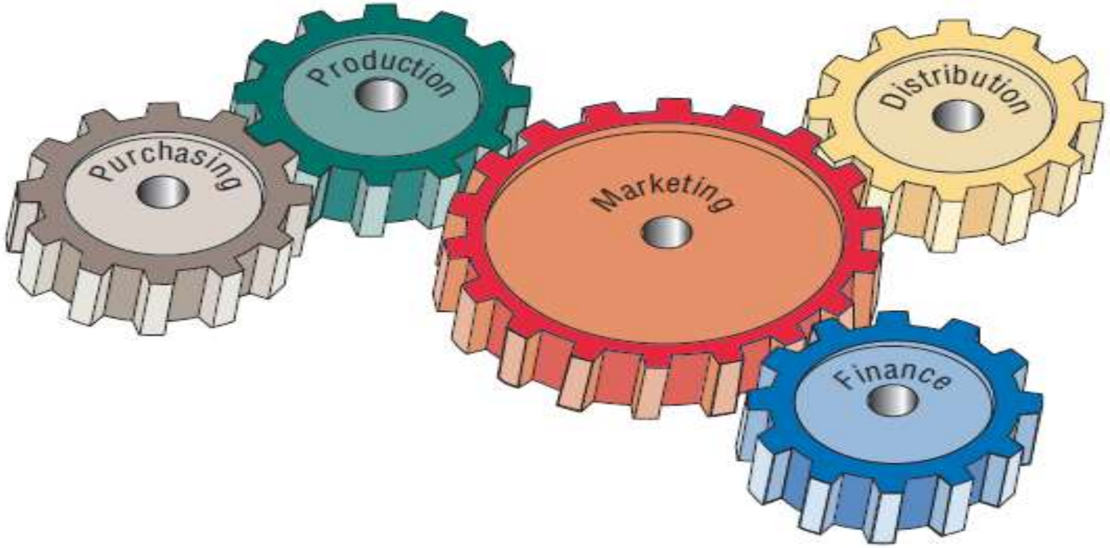
# Taking a Systems Perspective (Figure 2.2)

- Figure 2.2 shows the interrelated nature of systems
- Outputs from one department serve as inputs for another such
- Subsystems are interrelated

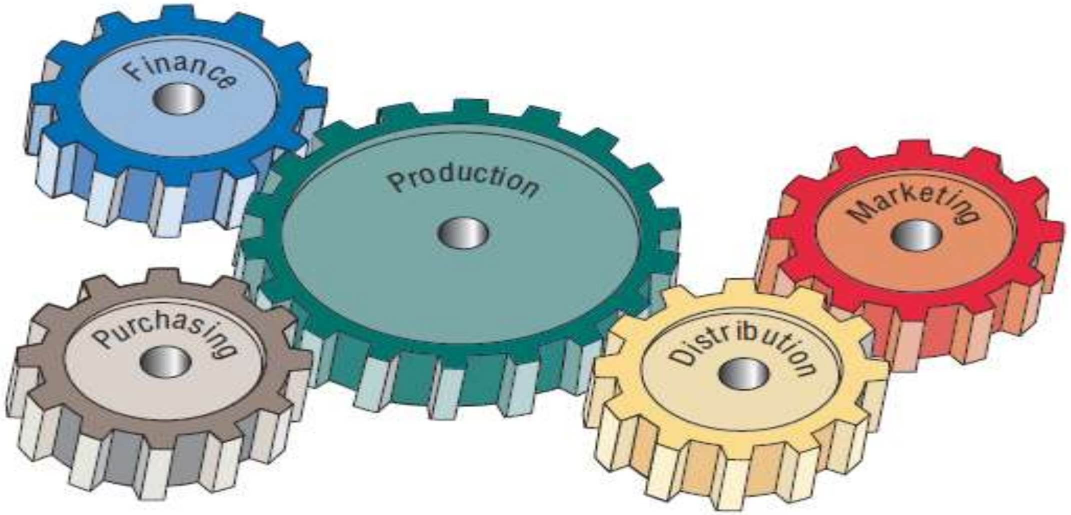

Outputs from marketing become the inputs for production.
Marketing
Production
Outputs from production become the inputs for marketing.

# Perspective of Functional Managers (Figure 2.3)

- In Figure 2.3 you can see:
  - The marketing and production managers view
  - The marketing manager’s personal  perspective shows the business as driven by marketing
  - All other functional areas are interrelated but are not of central importance



How a Marketing Manager May View the Organization



How a Production Manager May See the Organization

# Taking a Systems Perspective

- The marketing manager has the *marketing* subsystem as the focus
- Similarly, the perspective of a production manager focuses on the *production* subsystem along with all other functional areas driven by the production function
  - The relative importance of functional areas (as revealed in the personal perspectives of managers) takes on added significance when managers become strategic managers
  - They can create problems if they overemphasize their prior functional information requirements in relation to the broader needs of the organization

# Enterprise Systems: Viewing the Organization as a System

# Enterprise Resource Planning

- Enterprise system, or enterprise resource planning (ERP) system
  - Describes an integrated organization (enterprise) information system.
- Specifically, ERP is software that helps the flow of information between the functional areas in the organization.
  - Rather than being developed in-house, the system is usually purchased from one of the software development companies well known for its ERP packages, such as SAP or Oracle.
  - The product is then customized to fit the requirements of a particular company.
- Typically, the vendor requires commitment from the organization in terms of specialized user or analyst training.
- Over the last few years, ERP systems are moving to cloud computing.

# Enterprise Resource Planning

- ERP has taken root in many large companies and is spreading to small and medium-sized enterprises as well
- Most of the lessons systems analysts have learned are applicable to implementing enterprise systems.
  - However, as an analyst you need to recognize and act on the similarities and differences between implementing networked information systems and implementing an ERP.
- One of the major differences is that:
  - Rather than redesigning business processes based on a logical analysis of those processes and how they support the business strategy, and then choosing the IT to support those processes
  - A large installation of ERP can reverse this by requiring the implementation of new business processes that are embedded in the technology provided
- An internal systems team must maintain the interface between legacy systems and the ERP being installed

# Enterprise Resource Planning

- Enterprise implementations are complex and intense endeavors that result in tremendous organization change affecting every aspect of the organization including the:
  - Design of employees' work
  - Skills required to become competent in one's job
  - strategic positioning of the company
- Implementing an ERP has become increasingly complex - many issues present important hurdles to clear if the ERP installation is to be declared a success including:
  - User acceptance
  - Integration with legacy systems
  - The supply chain
  - Upgrading functionality (and complexity) of ERP modules
  - Reorganizing work life of users and decision makers
  - ANALYSIS FUNDAMENTALS expanded reach across several organizations (as part of the IT supply chain)
  - strategic repositioning of the company adopting an ERP.

# Enterprise Resource Planning

- New research has shown that within 2 to 3 years following implementation:
  - ERP can result efficiencies
  - Making employees more  effective
- In the planning and early stages of ERP:
  - Deep analysis examines how the new enterprise system will affect the daily lives and jobs of employees involved in the new system (see chapter – HCI and usability issues)
- ERP systems are best considered:
  - As real game changers when it comes to  the job design of employees (including systems analysts) and as a moving force that can alter the organizations strategic approach

# Depicting Systems Graphically

# System Modelling

- A system or subsystem as it exists within a corporate organization may be graphically depicted in several ways
- The various graphical models show the boundaries of the system and the information used in the system
- The predominant approaches to modelling systems graphically are:
  - Context-level data flow diagrams
  - Entity-relationship model
  - Use case modeling
- These methods of modeling both existing systems and new systems are:
  - Well known and understood in the ‘real-world’
  - Have plentiful supporting software applications
  - Are supported in education and training

# Systems and the Context-Level Data Flow Diagram

# Data Flows

- §Focus is on the data flowing into and out of the system and the processing of the data
- Shows the scope of the system:
  - What is to be included in the system
  - The external entities are outside the scope of the system

# ?

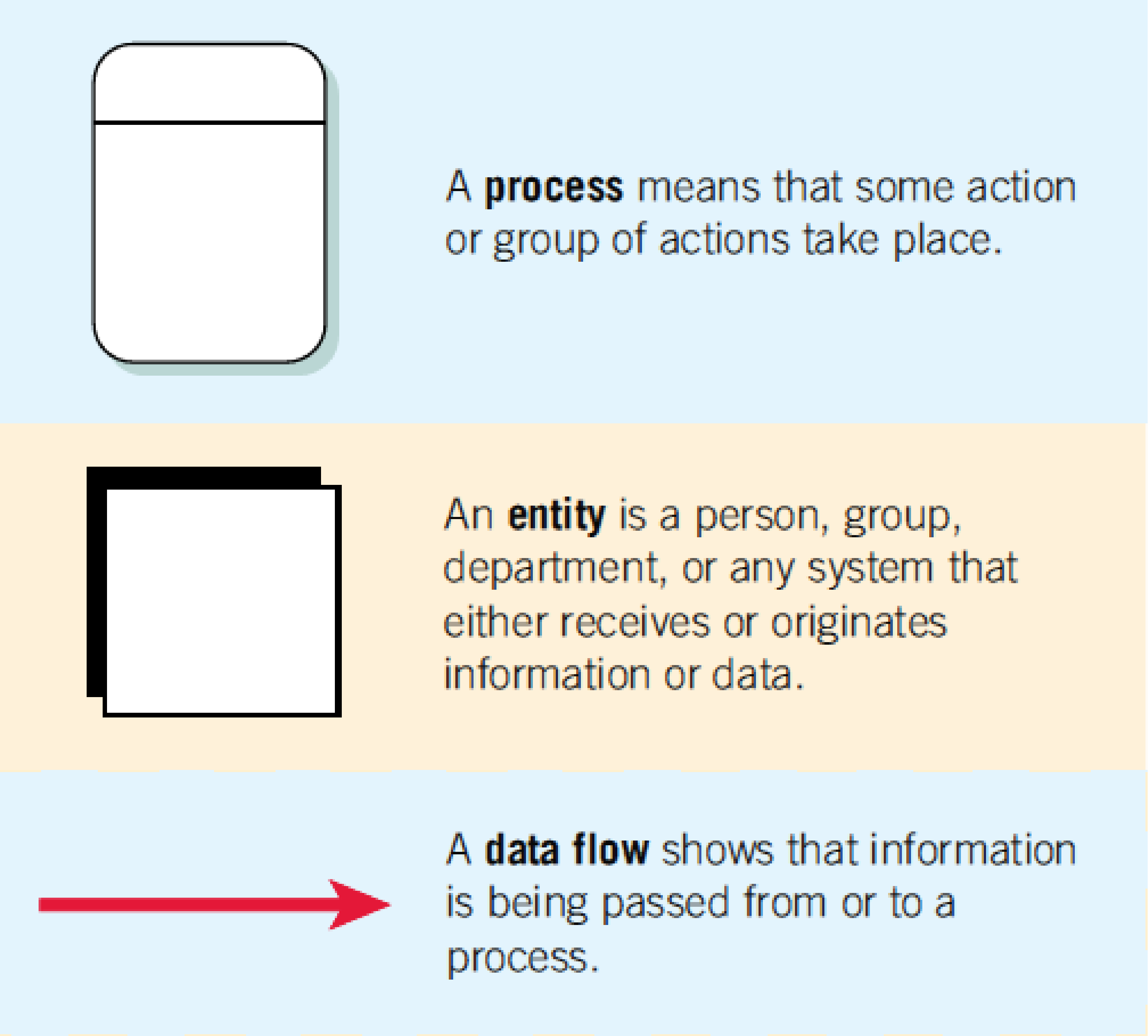
- The first model is the context-level data flow diagram
- Data flow diagrams focus on the data flowing into and out of the system and the processing of the data
- These basic components of every computer program can be described in detail and used to analyze a system for accuracy and completeness
- As shown in Figure 2.4 the context-level data flow diagram employs only three symbols:
  - A rectangle with rounded corners
  - A square with two shaded edges
  - An arrow

# Data Flow Diagram Legend

- In Figure 2.4:
  - Processes transform incoming data into outgoing information and the content level has only one  process, representing the entire system
  - The external entity represents any entity that supplies or  receives information from the system but is not a part of the system
- An entity may be a:
  - *Person* / *a group of people* / *a corporate position or department* / or *other systems*
- The lines that connect the external entities to the process are called data flows and represent movement data
- The context-level data flow diagram serves as a  good starting point for drawing the use case diagram (discussed later in this chapter)

# Figure 2.4

- Processes transform incoming data into outgoing information and the content level has only one  process, representing the entire system
- The external entity represents any entity that supplies or  receives information from the system but is not a part of the system



A **process** means that some action or group of actions take place.

An **entity** is a person, group, department, or any system that either receives or originates information or data.

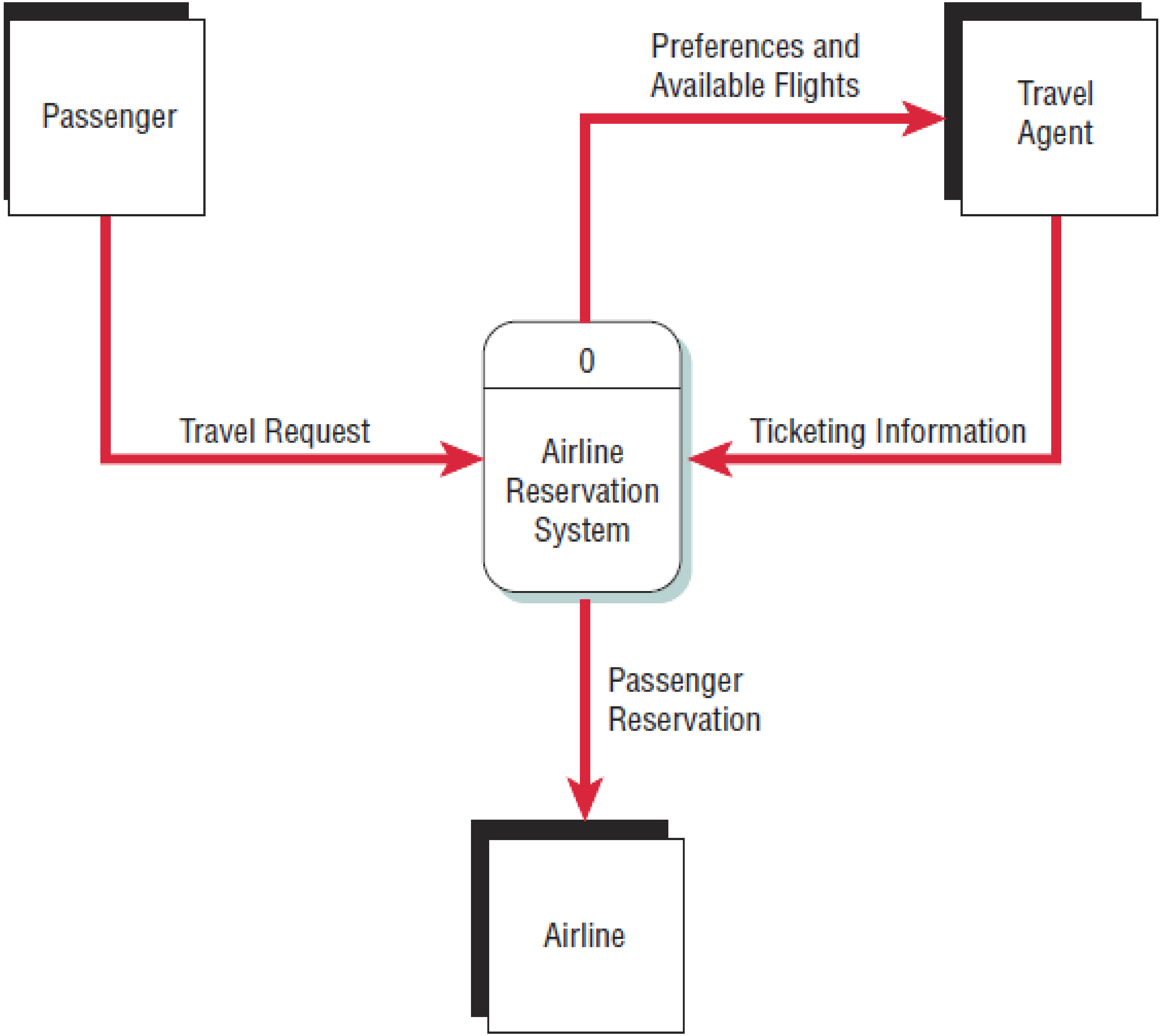
A **data flow** shows that information is being passed from or to a process.

# An Example Context-Level Data Flow Diagram

- An example of a context-level data flow diagram is found in Figure 2.5
  - This example represents the most basic elements of an airline reservation system and shows the system before (when a travel agent acted as intermediary) and after direct online booking was made possible.
  - In  the diagram
    - The passenger (an entity) initiates a travel request (the data flow)
- The context level diagram doesn’t show enough detail to indicate exactly what happens (it isn’t supposed  to) but:
  - We can see that the passenger’s preferences and the available flights are sent to the travel agent which sends ticketing information back to the process
- We can also see that the passenger reservation is sent to the airline.
  - The second context-level diagram (see the book pp. 63) shows what happens when the passenger books directly with an online system

# Figure 2.5

- This example represents the most basic elements of an airline reservation system and shows the system before (when a travel agent acted as intermediary) and after direct online booking was made possible.
- In  the diagram the passenger (an entity) initiates a travel request (the data flow)



# ?

- In Chapter 7 we will see that A data flow contains much information – for example:
  - The  passenger reservation contains the:
    - Passenger’s name / airline / flight number(s) / date(s) of travel / price / seating preference / meal preferences/ etc
  - For now we are concerned mainly with how a  context-level data flow diagram defines the boundaries of the system
- In the preceding example:
  - Only reservations are part of the process
  - Other decisions that the airline would make (for  example, purchasing airplanes, changing schedules, pricing, and hiring additional flight crew  members) are not part of this system
  - The context-level data flow diagram is one way to show:
    - The scope of the system or what  is to be included in the system
    - The external entities are outside the scope
    - Elements over  which the system has no control

# Systems and the Entity-Relationship Model

# Entity-Relationship Modelling

- An alternative method to model the scope of a system and define proper system boundaries is to use an entity-relationship (ER) diagrams (models)
- The focus is on elements that make up an organizational system can be referred to as entities and show a systems scope
  - An entity may be a person, a place, or a thing, such as a passenger on an airline, a destination, or a plane.
  - Alternatively, an entity may be an event, such as the end of the month, a sales period, or a machine breakdown.
- A relationship is the association that describes the interaction among the entities

# Entity-Relationship Modelling

- There are a number of conventions used in drawing ER diagrams
  - There are names such as crow's foot / arrow / or Bachman notation
  - These naming conventions (there are alternative notations used) and related names are well known and understood in the industry
- Relationships show how the entities are connected
- There are three types of relationships:
  - One-to-one
  - One-to-many
  - Many-to-many

# Entity-Relationship Modelling

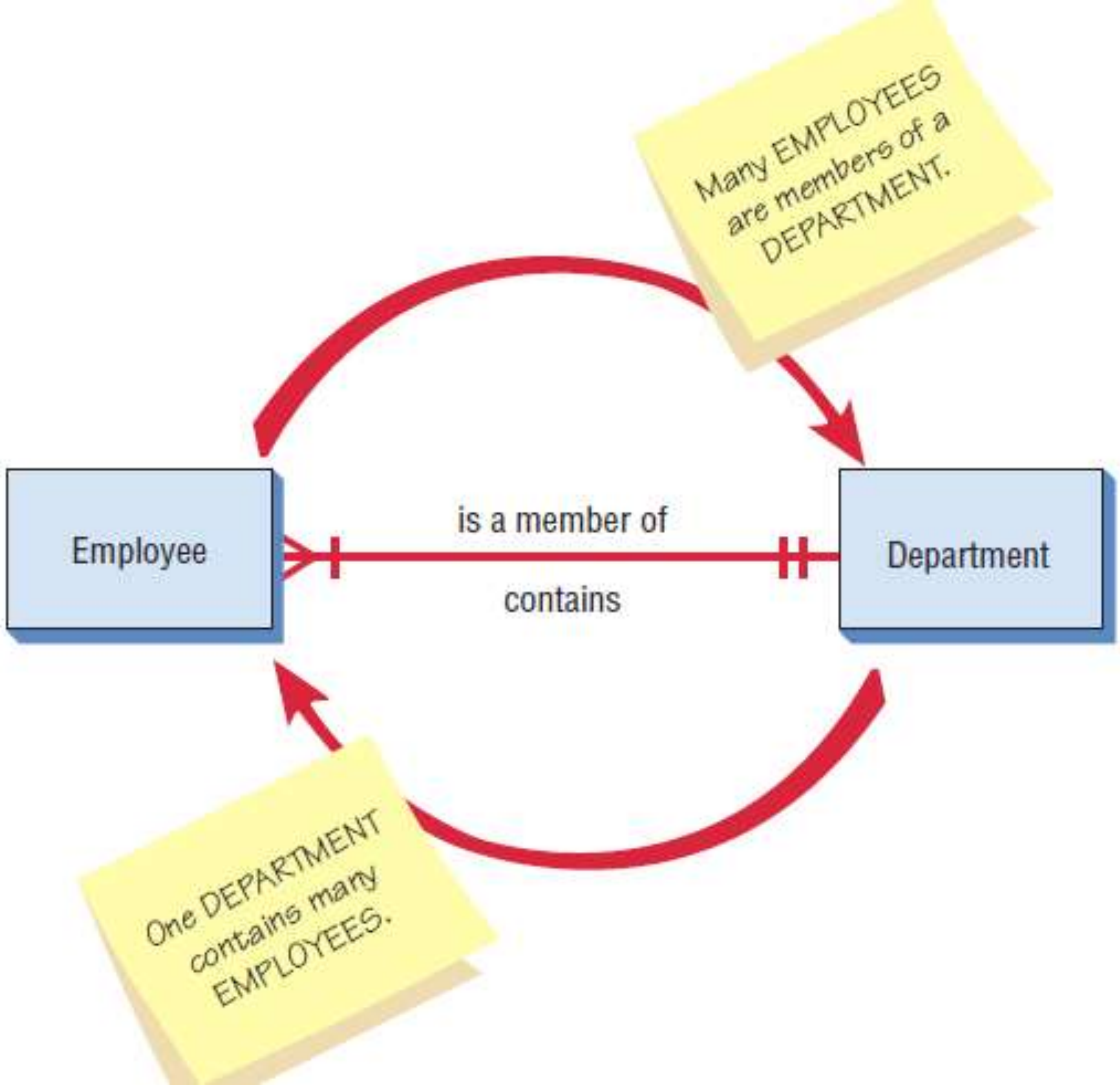
- There are a number of conventions used in drawing ER diagrams
  - There are names such as crow's  foot / arrow / or Bachman notation
  - These naming conventions (there are alternative notations used) and related names are well known and understood in the industry
- In this book:
  - The crow's foot notation is used to identify the relationship type
  - We assume  that an entity is a plain rectangular box
- Figure 2.6 illustrates a simple E-R diagram where:
  - Two entities are linked together by a line
  - In  this example: the end of the line is marked with two short parallel marks (| |) signifying that this *relationship* is *one-to-one*
  - Thus: exactly one employee is assigned to one phone extension
  - No one shares the same phone extension in this office

# Entity-Relationship Modelling

- The red arrows are not part of the E-R diagram; they are present to demonstrate how to read the diagram
  - The phrase on the right side of the line is read from top to bottom as follows: “One EMPLOYEE is assigned to one PHONE EXTENSION”
  - On the left side, as you read from bottom to top, the arrow says, “One PHONE EXTENSION is listed for one EMPLOYEE”
- Figure 2.7 shows another relationship:
  - The crow’s foot notation (**>—+**) is obvious on this diagram and this particular example is a ***many-to-one*** example
  - As you read from left to right, the arrow signifies, “Many EMPLOYEES are members of a DEPARTMENT”
  - As you read from right to left, it implies, “One DEPARTMENT contains many EMPLOYEES”

# ER Diagram

- Entity-Relationship Example (Figure 2.7)
- Figure 2.7 shows an entity-relationship diagram
- Shown is a ***many-to-one*** relationship
- As you read from left to right, the arrow signifies, “Many EMPLOYEES are members of a DEPARTMENT”
- As you read from  right to left, it implies, “One DEPARTMENT contains many EMPLOYEES”

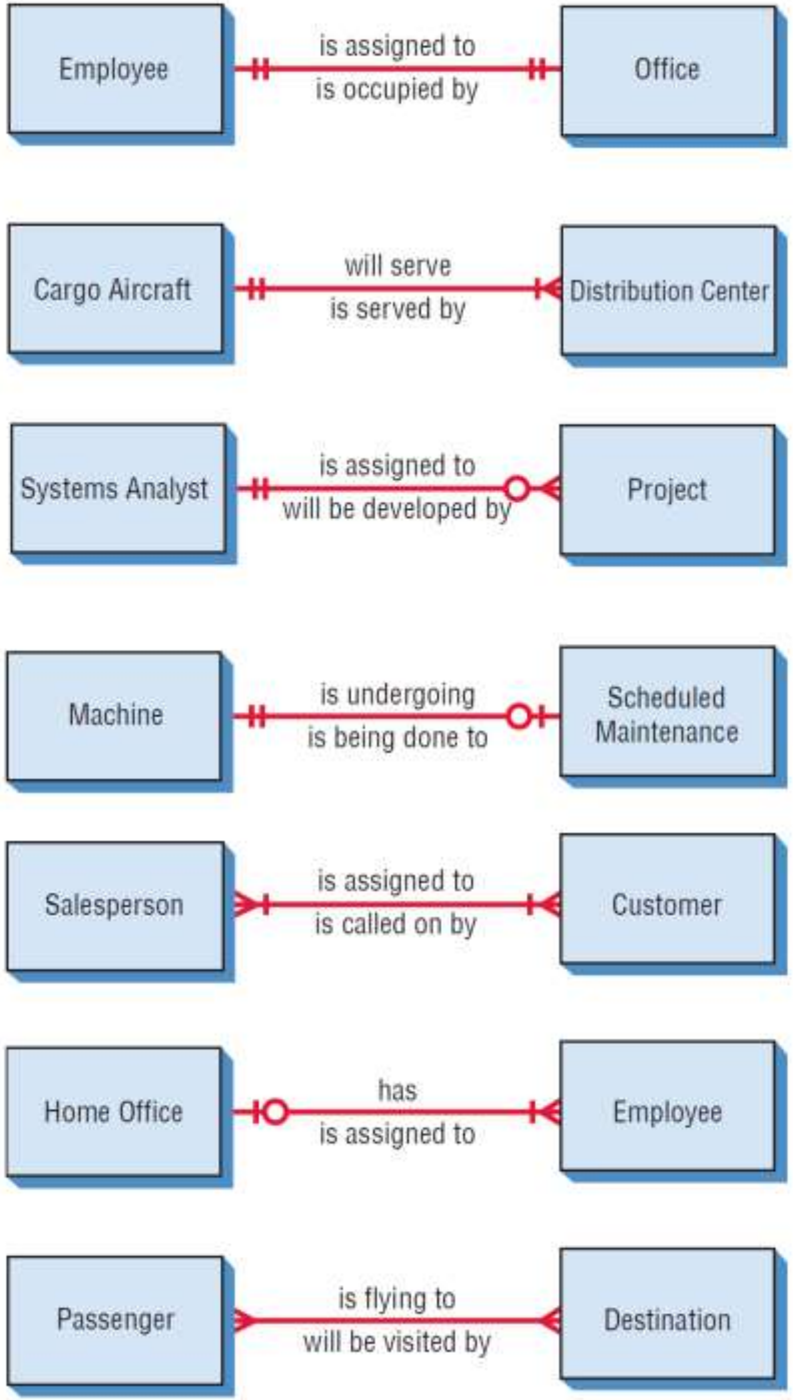

Many EMPLOYEES are members of a DEPARTMENT.
Employee
is a member of
contains
Department
One DEPARTMENT contains many EMPLOYEES.

# Entity Relationship Modelling

- In Figure 2.7 notice that:
  - When a ***many-to-one*** relationship is present
  - The grammar changes from "*is*" to "*are*" even though the singular "*is*" is written on the line.
  - The crow's foot and the single mark does not literally mean that this end of the relationship must be a mandatory "*many*."
    - They imply that this end could be anything from *one* to *many*
- Figure 2.8 shows a number of different ER modelling relationships

# Figure 2.8

- Figure 2.8 elaborates on Figure 2.7
- Shown are examples of different types of relationships in ER diagrams
- Here we have listed:
  - A number of potential entities in a typical system
  - We show a number of typical entity relationships between the entities
  - The crows-foot notation is clearly shown



# ?

- Figure 2.8 elaborates on this scheme
  - Here we have listed a number of typical entity relationships
  - The first is a one-to-one relationship:
    - "An EMPLOYEE is assigned to an OFFICE."
  - The second is a one-to-many relationship:
    - "One CARGO AIRCRAFT will serve one or more DISTRIBUTION CENTERs."
  - The third one is slightly different because it has a circle at one end
    - It can be read as "A SYSTEMS ANALYST may be assigned to MANY PROJECTs," meaning that the analyst can be assigned to no projects [that is what the circle (O), for zero, is for], one project, or many projects.
    - Likewise, the circle (O) indicates that none is possible in the next relationship. Recall that the short mark means one.
    - Therefore, we can read it as follows: "A MACHINE may or may not be undergoing SCHEDULED MAINTENANCE."
  - Notice that the line is written as "is undergoing," but the end marks on the line indicate that either no maintenance (O) or maintenance (I) is actually in progress

# ?

- The next relationship states:
  - "One or many SALESPEOPLE" (plural of SALESPERSON) are assigned to one or more CUSTOMERs."
  - It is the classic many-to-many relationship.
- The next relationship can be read as follows:
  - "The HOME OFFICE can have one or many EMPLOYEEs" or "One or more EMPLOYEEs may or may not be assigned to the HOME OFFICE"
  - Once again, the I and O together imply a Boolean situation (In other words: one or zero)
- The final relationship shown can be read as:
  - "Many PASSENGERs are flying to many DESTINATIONs"
  - Some analysts prefer this symbol [**>—+**] to indicate a mandatory "many" condition
  - However: would it ever be possible to have only one passenger or only one destination?
  - Even so:, some CASE tools such as Visible Analyst do not offer this possibility because the optional *one-or-many* condition as shown in the SALESPERSON–CUSTOMER relationship is an adequate notation

# ?

- Up to now we have modeled all our relationships using one simple rectangle and a line
  - This method works well when we are examining the relationships of real things such as real people, places, and things
  - Sometimes, though, we create new items in the process of developing an information system.
  - Some examples are *invoices*, *receipts*, *files*, and *databases*
  - When we want to describe how a person relates to a receipt, for example, it becomes convenient to indicate the receipt in a different way.

# ?

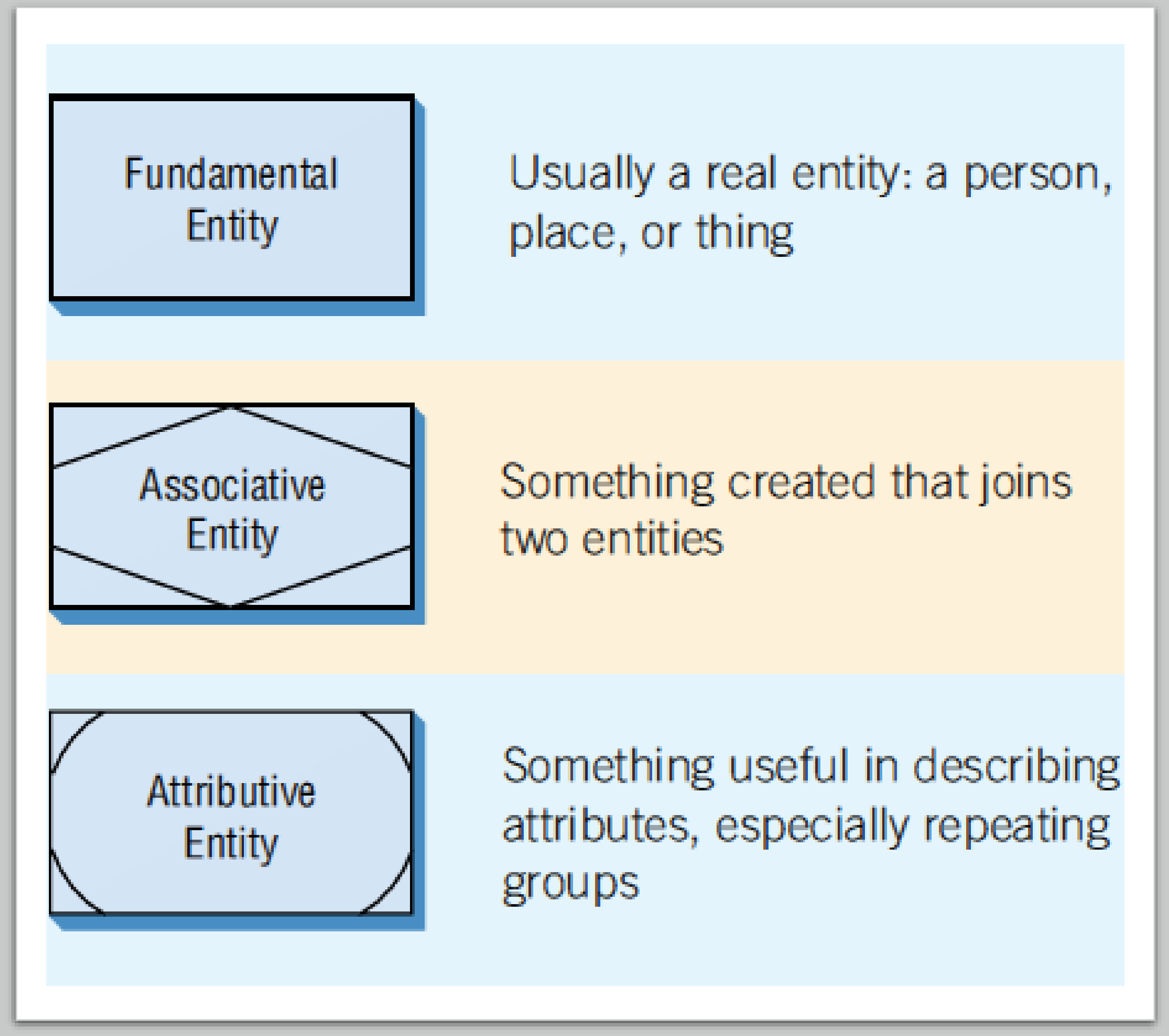
- There are three different types of entities:
    - The *fundamental* entity
    - The *associative* entity
    - The *attributive* entity
- The three entity types are shown in Figure 2.9
- An associative entity can exist only if it is connected to at least two other entities – for that reason it may be called:
    - A *gerund*
    - A *junction*
    - An *intersection*
    - A *concatenated entity*
- This wording makes sense because a receipt wouldn’t be necessary unless there were a customer and a salesperson making the transaction

# ?

- Another type of entity is an *attributive* entity
- When an analyst wants to show data that are completely dependent on the existence of a fundamental entity an attributive entity should be used.
    - For example: If a library has multiple copies of the same book
        - An attributive entity can be used to designate which copy of the book is being checked out
- The *attributive* entity is useful for showing repeating groups of data
    - For example: Suppose we are going to model a relationships that exist when a patron gets tickets to a concert or a show:
    - The entities seem obvious at first:
        - “a PATRON and a CONCERT/SHOW” (as shown in Figure 2.10)

# Figure 2.9

- Three Different Types of Entities Used in E-R Diagrams



Usually a real entity: a person, place, or thing

Something created that joins two entities

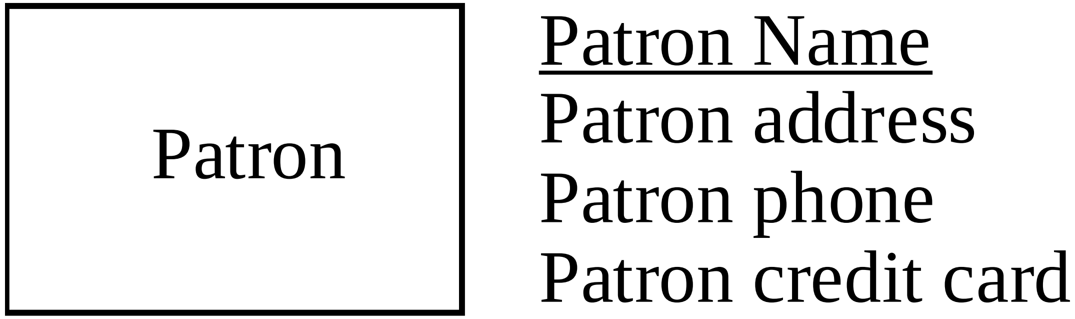
Something useful in describing attributes, especially repeating groups

# ?

- For the Figures 2.10 and 2.11 refer to the text-book:
  - 2.10 (pp. 66)
    - The first attempt at drawing an ER diagram
  - 2.11 (pp. 67)
    - Improving the ER diagram by  adding an associative entry called “RESERVATION”
  - FIGURE 2.12 (pp. 68)
    - Is a more complete E-R diagram showing data attributes of the entities

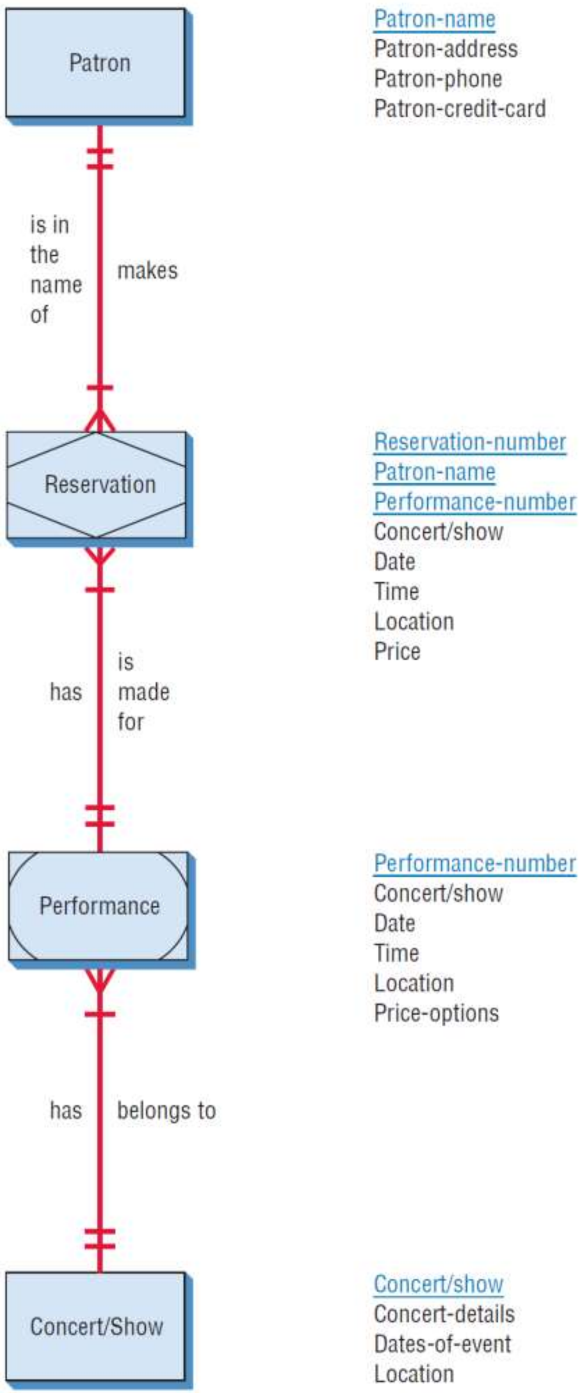
?

- A More Complete E-R Diagram Showing Data Attributes of the Entities (Figure 2.12 )
- Data attributes may be added to the diagram.



Patron

<u>Patron Name</u>
Patron address
Patron phone
Patron credit card



# ?

- At first glance, we see that the PATRON gets a reservation for a CONCERT/SHOW, and the CONCERT/SHOW can be said to have made a booking for a PATRON.
- The process isn’t that simple, of course, and the E-R diagram need not be that simple either.
- The PATRON actually makes a RESERVATION, as shown in Figure 2.11. The RESERVATION is for a CONCERT/SHOW.
- The CONCERT/SHOW holds the RESERVATION, and the RESERVATION is in the name of the PATRON.
- We added an associative entity here because a RESERVATION was created due to the information system required to relate the PATRON and the CONCERT/SHOW.

# ?

- Again this process is quite simple, but because concerts and shows have many performances, the E-R diagram is drawn once more in Figure 2.12.
- Here we add an attributive entity to handle the many performances of the CONCERT/SHOW. In this case the RESERVATION is made for a particular PERFORMANCE, and the PERFORMANCE is one of many that belong to a specific CONCERT/SHOW.
- In turn, the CONCERT/SHOW has many performances, and one PERFORMANCE has a RESERVATION that is in the name of a particular PATRON.
- To the right of this E-R diagram is a set of data attributes that make up each of the entities. Some entities may have attributes in common.
- The attributes that are underlined can be searched for. The attributes are referred to as keys and are discussed in Chapter 13.

# ?

- Systems designers often use E-R diagrams to help model a file or a database. It is even more important, however, that a systems analyst understand both the entities and relationships in the organizational system very early.
- In sketching out some basic E-R diagrams, the analyst needs to:
    - 1. List the entities in the organization to gain a better understanding of the organization.
    - 2. Choose key entities to narrow the scope of the problem to a manageable and meaningful dimension.
    - 3. Identify what the primary entity should be.
    - 4. Confirm the results of steps 1 through 3 through other data-gathering methods (investigation, interviewing, administering questionnaires, observation, and prototyping), as discussed in Chapters 4 through 6.

# ?

- It is critical that a systems analyst begin to draw E-R diagrams upon entering an organization rather than waiting until the database needs to be designed.
- These E-R diagrams help the analyst understand what business the organization is actually in, determine the size and scope of the problem, and discern whether the right problem is being addressed.
- The E-R diagrams need to be confirmed or revised as the data-gathering process takes place.

# Summary

- This Chapter has provided an overview and introduction to the creation of entity-relationship diagrams
- In summary there are four essential steps:
  - Analyze the organization and culture
  - List the entities in the organization
  - Identify the key entities (to narrow the scope of the problem)
  - Identify the primary entity
  - Confirm the results of the above through data gathering

# Use Case Modelling

# Use-Case Models in UML

- The *Unified Modeling Language* (UML):
  - Has been adopted as a standard by the *Object Management Group* (OMG)
  - Familiarity with UML by software engineers is now ubiquitous
  - UML is an industry standard for modelling and system design
- Use-Case modelling:
  - Originally introduced as a diagram for use in object-oriented unified modeling language (UML)
  - Use cases are now being used regardless of the approach to systems development.

# Use-Case Models in UML

- A use case  diagram can be used as part of the systems development life cycle (SDLC) approach or in agile  modeling
  - The word use is pronounced as a noun (“yoos”) rather than a verb (“yooz”)
- A use case  model describes what a system does without describing how the system does it:
  - It is a  logical model of the system
    - Logical and conceptual models are further discussed in Chapter 7
  - A use case model:
    - Reflects the view of a system from the perspective of a user outside the system (the system requirements)

# Use-Case Models in UML

- An analyst develops use cases in a cooperative effort with the business experts who help define the requirements of the system
- A use case model provides:
  - An effective means of communication between the business team and the development team
- A use case model:
  - Partitions the way the system works into:
  - Behaviors / services / responses
  - The use cases that are significant for the system design and to the users of the system

# Use-Case Models in UML

- From the perspective of an actor (or user)
  - A use case should produce something of value
  - The analyst must determine what is important to the user and remember to include it in the use case diagram
- For example:
  - Is entering a username and password something of value to the user?
  - It may be included if the user has a concern about security or if it is critical to the success of the project

# Use Case Symbols

# Use-Case Symbols

- A use case diagram contains:
  - The actor (or multiple actors / multiple use of the same actor)
  - Symbols
  - Connecting lines
- Actors are similar to external entities as they exist outside the system
- The term actor refers to a particular role of a user of the system – for example:
  - An actor may be an employee but also may be a customer at the company store (dual roles)
  - Even though it is the same person in the real world it is represented as two different symbols on a use case diagram because the person interacts with the system in different roles.

# Use-Case Symbols

- Actors exists outside the system and interact with the system in a specific way
- An actor can be: a *human*, *another system*, or a *device such as a keyboard or a Web connection*.
- Actors can initiate an instance of a use case
- An actor may interact with one or more use cases and a use case may involve one or more actors
- Actors may be divided into two groups:
  - Primary actors supply data or receive information from the system
  - Some users directly interact with the system (system actors), but primary actors also may be businesspeople who do not directly interact with the system but have a stake in it

# Use-Case Symbols

- Actors exists outside the system and interact with the system in a specific  way
- An actor can be: a *human*, *another system*, or a *device such as a keyboard or a Web connection*.
- Actors can initiate an instance of a use case
- An actor may interact with one or more use  cases and a use case may involve one or more actors
- Actors may be divided into two groups:
    - Primary actors
    - Secondary actors

# Primary Actors

- Primary actors
  - Primary actors supply data or receive information  from the system
  - Some users directly interact with the system (system actors), but primary actors  also may be businesspeople who do not directly interact with the system but have a stake in it
- Primary actors are important because:
  - They are the people who use the system
  - Can provide  details on what the use-case should do
  - They also can provide a list of goals and priorities

# Secondary Actors

- Supporting actors (also called secondary actors) help to keep the system running or provide other services
  - Supporting actors are the analysts, programmers, and people who run the help desk, etc
- Sometimes it is useful to create an actor profile that lists the actors, their background, and their skills in a simple table format
  - This may be useful to understand how the actor interacts with the system.
  - An example is an Order Processing Specialist.
    - The profile would be:
    - "A routine user of the software, familiar with minor features, order exceptions, and order customization."
  - It is also useful to list the actors and their goals and priorities
  - Each goal may become a use case.

# Use-Cases

- A use case provides developers with a view useful to users
  - It is free of technical or implementation details
  - We can think of a use case as a sequence of transactions in a system
- The use case model is based on the interactions and relationships of individual use cases
- A use case always describes three things:
  - An actor that initiates an event
  - The event that triggers a use case, and the use case that performs the actions triggered by the event
  - In a use case an actor using the system initiates an event that begins a related series of interactions in the system
- A use case documents a single transaction or event

# Use-Cases

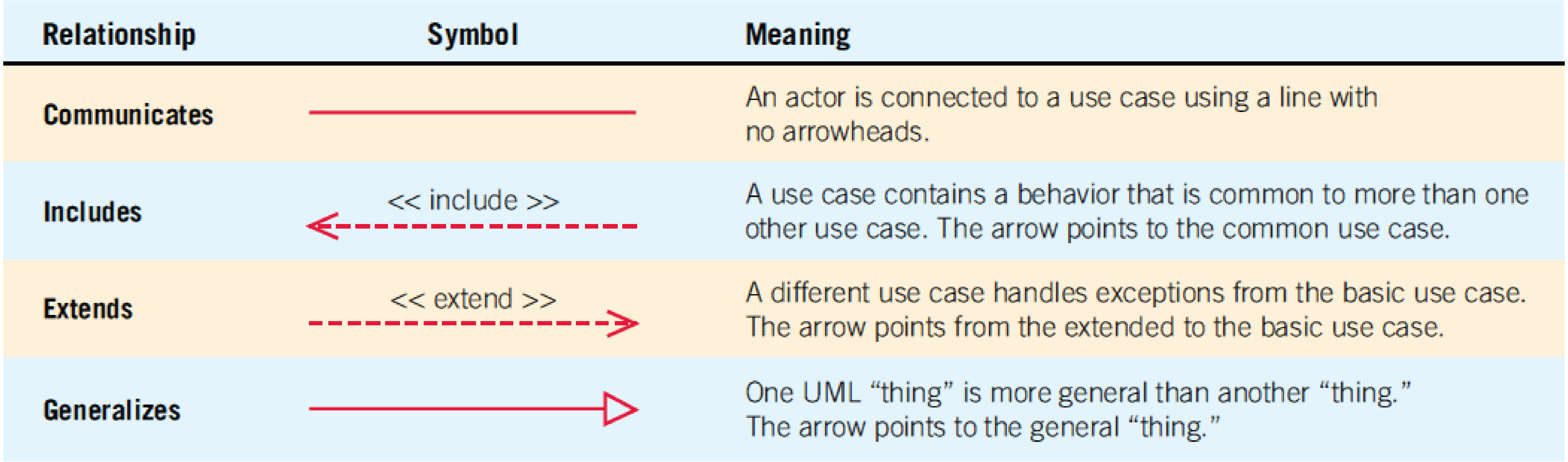
- An event is an input to the system that happens at a specific time and place and causes the system to do something
- It is better to create fewer use cases rather than more
- Often queries and reports are not  included:
  - 20 use cases (and no more than 40 or 50) are generally sufficient for a large system
- Use cases  also may be nested as required
- Some use cases use the verb manage to group use cases for:
  - adding / deleting / changing into another lower-level use case diagram
- You can include a use  case on several diagrams:
  - But the actual use case is defined only once in the repository
- A use  case is named with a verb and a noun

# Use Case Relationships

# Summary

- Active relationships are referred to as behavioral relationships and are used primarily in use case diagrams
- There are four basic types of behavioral relationships:
  - COMMUNICATES
  - INCLUDES
  - EXTENDS
  - GENERALIZES
- Notice that all the terms are action verbs
- Figure 2.13 shows the relationships and arrows and lines used in the UML diagram for each of the four types of behavioral relationships
- The four relationships are described next.

# Behavioral Relationships (Figure 2.13)



| Relationship | Symbol | Meaning |
|---|---|---|
| **Communicates** | ——— | An actor is connected to a use case using a line with no arrowheads. |
| **Includes** | << include >><br>← - - - - - - | A use case contains a behavior that is common to more than one other use case. The arrow points to the common use case. |
| **Extends** | << extend >><br>- - - - - - → | A different use case handles exceptions from the basic use case. The arrow points from the extended to the basic use case. |
| **Generalizes** | ———▷ | One UML "thing" is more general than another "thing." The arrow points to the general "thing." |

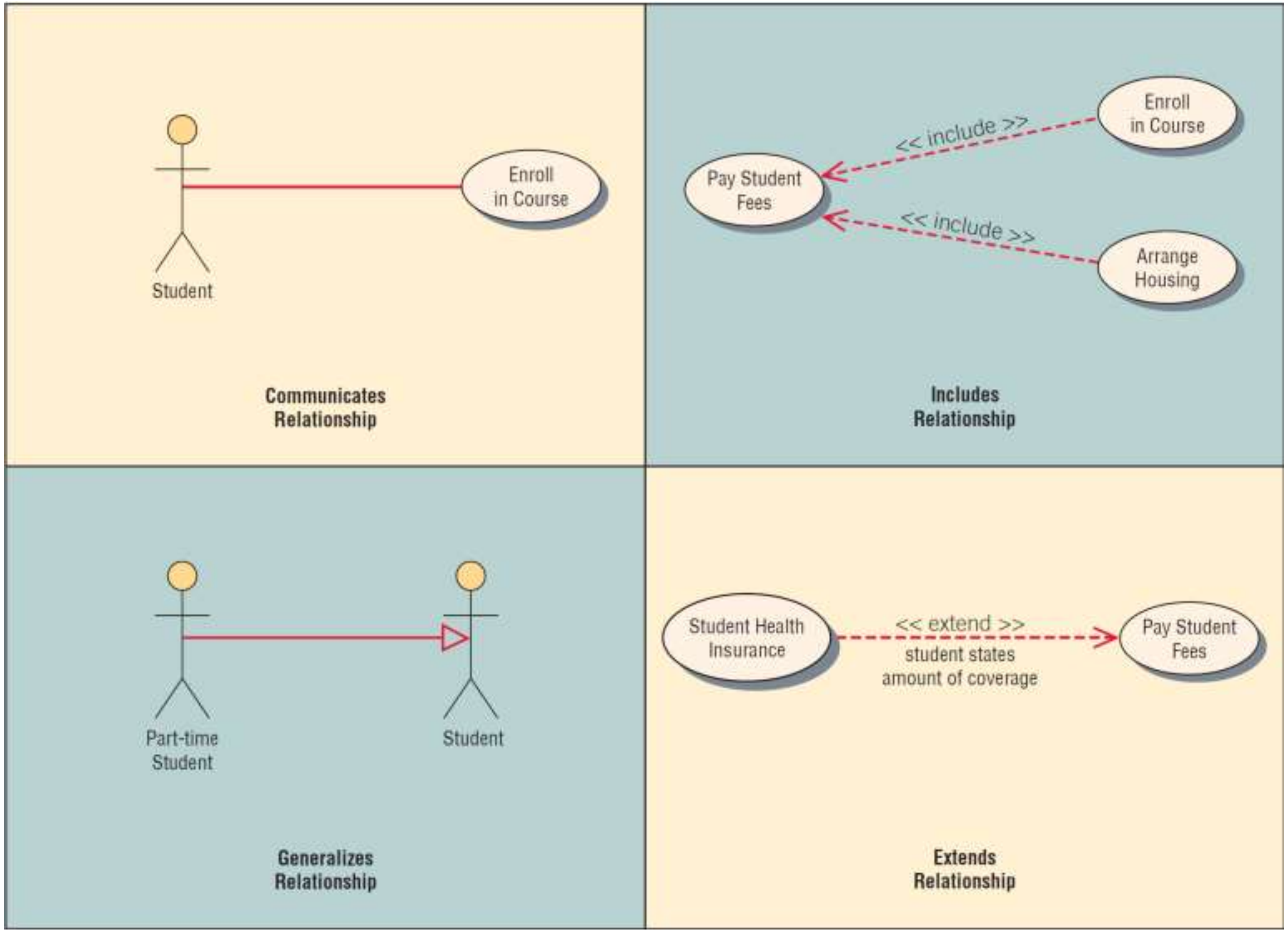
Figure 2.13 shows the  arrows and lines used in the UML diagram for each of the four types of behavioral relationships

# Communicates Relationship

- COMMUNICATES:
- The behavioral relationship communicates is used to connect an actor to a use case
  - Remember that the task of a use case is to give some sort of result that is beneficial to the actor in the system
  - Therefore, it is important to document these relationships between actors and use cases
- In our first example, a “Student” communicates with “Enroll” in Course
  - Examples of some components of a student enrollment example are shown in the use case diagrams in Figure 2.14

# Figure 2.14

- Some components of use case diagrams showing actors, use cases, and relationships for a student enrollment example
- Examples of some components of a student enrollment example are shown in the use case diagrams



# Includes Relationship

- INCLUDES:
- Also called the uses relationship this describes the situation in which a use case contains behavior that is common to more than one use case
  - Where the common use case is included in the other use cases
  - A dotted arrow that points to the common use case indicates the includes relationship
- An example would be a use case Pay Student Fees that is included in  Enroll in Course and Arrange Housing because in both cases students must pay their fees.
- Pay Student Fees may be used by several use cases
  - The arrow points toward the common use case

# Extends Relationship

- EXTENDS:
  - This relationship describes the situation in which one use case possesses the behavior that allows the new use case to handle a variation or an exception from the basic use case
- For example:
  - The extended use case Student Health Insurance extends the basic use case Pay Student Fees
  - The arrow goes from the extended to the basic use case.

# Generalizes Relationship

- GENERALIZES:
  - This relationship implies that one thing is more typical than the other thing
  - This relationship may exist between two actors or two use cases
- For example:
  - A part-time  Student generalizes a Student
  - Similarly some of the university employees are professors
- The  arrow points to the general thing.

# Developing System Scope

# System Scope

- The scope of a system defines its boundaries:
  - What is within the scope of a system (inside the system)
  - What is external to the system (outside the system – out of scope
- Projects will have a budgetary plan that helps to define:
  - The scope of the project
  - The time scales (the start and end dates)
- Actors are always outside the scope of the system
- The COMMUNICATES lines that connect actors to the use cases are the boundaries and define the scope
- Because a use case diagram  is created early in the system’s life cycle:
  - the budget / start and end dates may (in practice always will) change as the project progresses
  - As the analyst learns more about the system the use case diagrams, use  case(s), and system (project) scope may change

# Developing Use-Case Diagrams

# Developing Use Case Diagrams

- The primary use case:
    - Consists of a standard flow of events in the system that describes a standard system behavior
    - Represents the normal, expected, and successful completion of the use case
- When diagramming a use case:
    - Start by asking the users to list everything the system should do for them
        - This can be done using interviews in a joint application design session (as described in Chapter 4) (or)
        - Through other facilitated team sessions
    - The analyst may also use agile stories sessions (described in Chapter 6) to develop use cases
- Write down who is involved with each use case and the responsibilities or services the use case must provide to actors or other systems
- In the initial phases, this may be a partial list that is expanded in the later analysis phases

# Developing Use Case Diagrams

- Use  the following guidelines:
  1. Review the business specifications and identify the actors involved
  2. Identify the high-level events and develop the primary use cases that describe those events  and how the actors initiate them
     - Carefully examine the roles played by the actors to identify all the possible primary use cases initiated by each actor. Use cases with little or no  user interaction do not have to be shown.
  3. Review each primary use case to determine the possible variations of flow through the use  case
     - From this analysis, establish the alternative paths
     - Because the flow of events is usually different in each case
       - Look for activities that could succeed or fail
       - Also look for any  branches in the use case logic in which different outcomes are possible

# Developing Use Case Diagrams

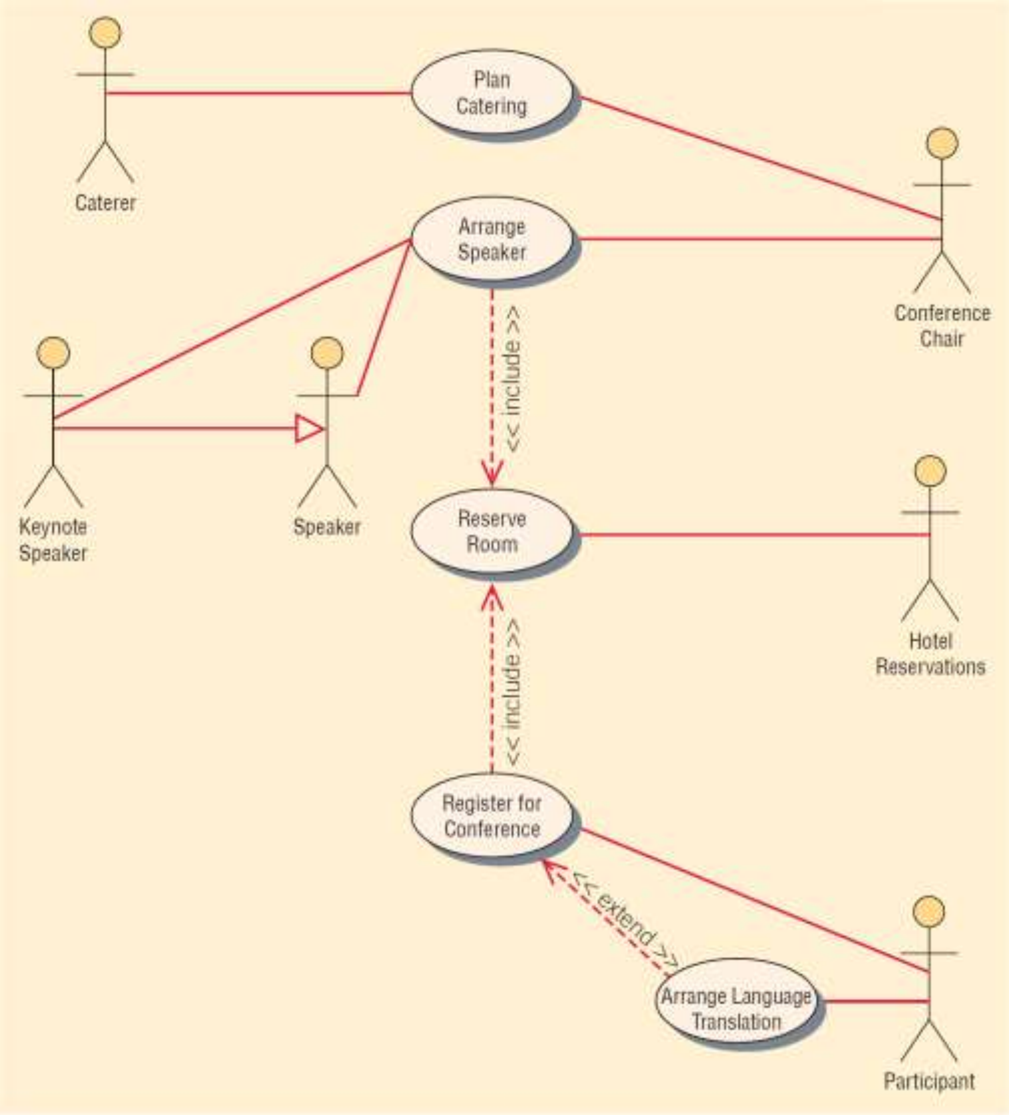
- If a context-level data flow diagram has been created it can be a starting point for creating a use case:
  - The external entities are potential actors
  - Then examine the data flow to determine whether it would initiate a use case or be produced by a use case
- Figure 2.15 is an example of a use case diagram representing a system used to plan a conference where the actors are the:
  - Conference Chair, responsible for planning and managing the  conference
  - Conference Participant
  - Speakers
  - Keynote Speaker
  - Hotel Reservations
  - Caterer

# Summary

- The roles
- Actors represent the role the user plays:
  - The *Caterer* may be either a hotel  employee or an external catering service.
  - Both the *Conference Chair* and the *Caterer* are involved in planning meals and banquets
  - The *Conference Chair* is also responsible for arranging *speakers*
  - The *Participant* registers for the conference

# Developing Use Case Diagrams (Figure. 2.15)

- A Use Case Diagram Representing a System Used to Plan a Conference
- Notice that the *Reserve Room* use case is involved in an ***includes*** relationship with the *Arrange Speaker* and *Register for Conference* use cases because both speakers  and participants will need lodging
- The *Arrange Language Translation* use case ***extends*** the *Register for Conference* use case because not all participants will require language translation  services.
- The Speaker actor is a ***generalization*** of Keynote Speaker

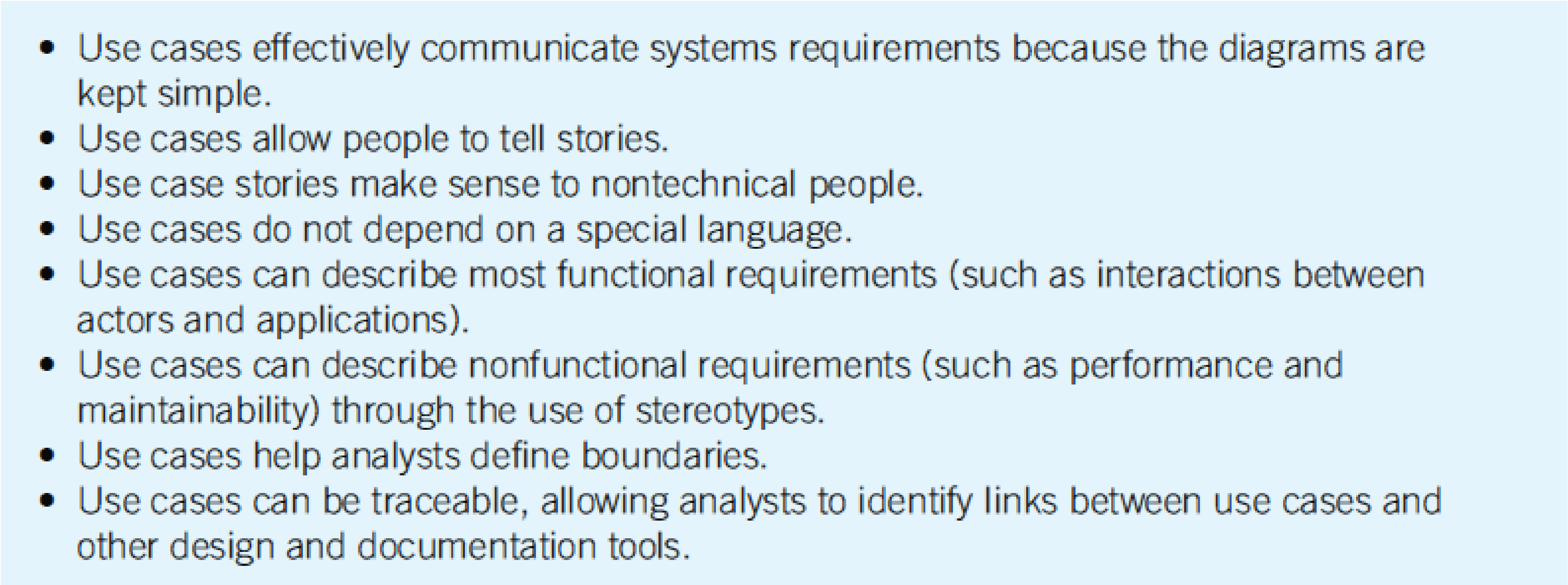


# Developing Use-Case Scenarios

# Developing Use Case Scenarios

- Each use case has a description:
    - We refer to the description as a use case scenario
- As noted earlier:
    - The primary use case represents the standard flow of events in the system, and alternative paths  describe variations to the behavior
    - Use case scenarios may describe what happens:
        - If an item purchased is out of stock (or)
        - If a credit card company rejects a customer’s requested purchase
- There is no standardized use case scenario format:
    - Each organization is faced with specifying what standards should be included
    - Often the use cases are documented using a template predetermined by the organization:
        - This approach can make use cases easier to read and provides standardized information for each use case in the model

# Writing Use Cases (Figure 2.18)



- Use cases effectively communicate systems requirements because the diagrams are kept simple.
- Use cases allow people to tell stories.
- Use case stories make sense to nontechnical people.
- Use cases do not depend on a special language.
- Use cases can describe most functional requirements (such as interactions between actors and applications).
- Use cases can describe nonfunctional requirements (such as performance and maintainability) through the use of stereotypes.
- Use cases help analysts define boundaries.
- Use cases can be traceable, allowing analysts to identify links between use cases and other design and documentation tools.

The Main Reasons for Writing Use Cases Are Their Effectiveness in Communicating with Users and Their Capturing of User Stories

# Use-Case Levels

# Use Case Levels

- You may want to create use cases for different levels
- One method uses 5 levels with the following altitude metaphors:
  1. *White* (is the highest level (e.g., clouds))
  2. *Kite* (lower than white) but still a high level providing an overview
  3. *Blue* (is at sea level) and is customarily used to depict user goals.
  4. *Indigo* (or fish) is a use case that shows lots of detail often at a functional or sub-functional level
  5. *Black* or *clam* (e.g., the bottom of the ocean) are the most detailed use cases at a subfunction level
- A use case scenario example is shown in Figure 2.16

# Use-Case Levels

- White is the highest level, like clouds.
    - This is the enterprise level, and there may be only  four to five use cases at this level for the entire organization. Examples might be to advertise goods, sell goods to customers, manage inventory, manage the supply chain, and optimize shipping.
- Kite is lower than white but still a high level, providing an overview.
    - The kite use case may  be at the business unit or department level and is a summary of goals. Examples would be  to register students, or, if working with a travel company, to make an airline, hotel, car, or  cruise reservation.
- Blue is at sea level and is customarily used to depict user goals.
    - This level often has the  greatest interest for users and is easiest for a business to understand. It is usually written  for a business activity, and each person should be able to do one blue level activity in anywhere from 2 to 20 minutes. Examples are register a continuing student, add a new customer, place an item in a shopping cart, and order checkout.

# Use-Case Levels

- Indigo or fish is a use case that shows lots of detail, often at a functional or sub-functional level
  - Examples are Choose a class, pay academic fees, look up the airport code for a given city, and produce a list of customers after entering a name
- Black or clam, like the bottom of the ocean, are the most detailed use cases, at a subfunction level
  - Examples might be to validate secure logon, add a new field using dynamic HTML, or use Ajax to update a web page in a small way
- Some of the areas included are optional and may not be used by all organizations. The three main areas are:
  1. An area header that contains case identifiers and initiators
  2. Steps performed
  3. A footer area that contains preconditions, assumptions, questions, and other information

# Use-Case Levels

- The method outlined hs been developed by Alistair Cockburn
- A full description may be found in the Chapter #2 (pp.72-76)
- There are figures which show the documentation:
  - FIGURE 2.16 A use case scenario is divided into three sections: identification and initiation; steps performed; and conditions, assumptions, and questions
  - FIGURE 2.17 Use cases can include conditional steps as well as extensions or alternative scenarios
- Read the text and study the figures

# Creating Use-Case Descriptions

# Creating Use-Case Descriptions

- § Use the following four steps to create use case descriptions:  1. Use agile stories, problem definition objectives, user requirements, or a features list as a  starting point.  2. Ask about the tasks that must be done to accomplish the transaction. Ask if the use case reads any data or updates any tables.  3. Find out if there are any iterative or looping actions.  4. The use case ends when the customer goal is complete.
- § FIGURE 2.18  Reasons for writing use cases.

# Why Use-Case Diagrams are Helpful

# Why Use Case Diagrams Are Helpful

- No matter what method you use to develop your system (traditional SDLC methods, agile methods, or object-oriented methods) you will find that use cases are valuable
- The use case diagrams identify all the actors in the problem domain
- A systems analyst can concentrate on:
  - What  people want and need to use the system
  - Extend their capabilities (and)
  - Enjoy their interaction with  technology
- The main reasons for writing use cases are shown in Figure 2.18 where the actions that need to be completed are also clearly shown on the use case diagram

# Why Use Case Diagrams Are Helpful

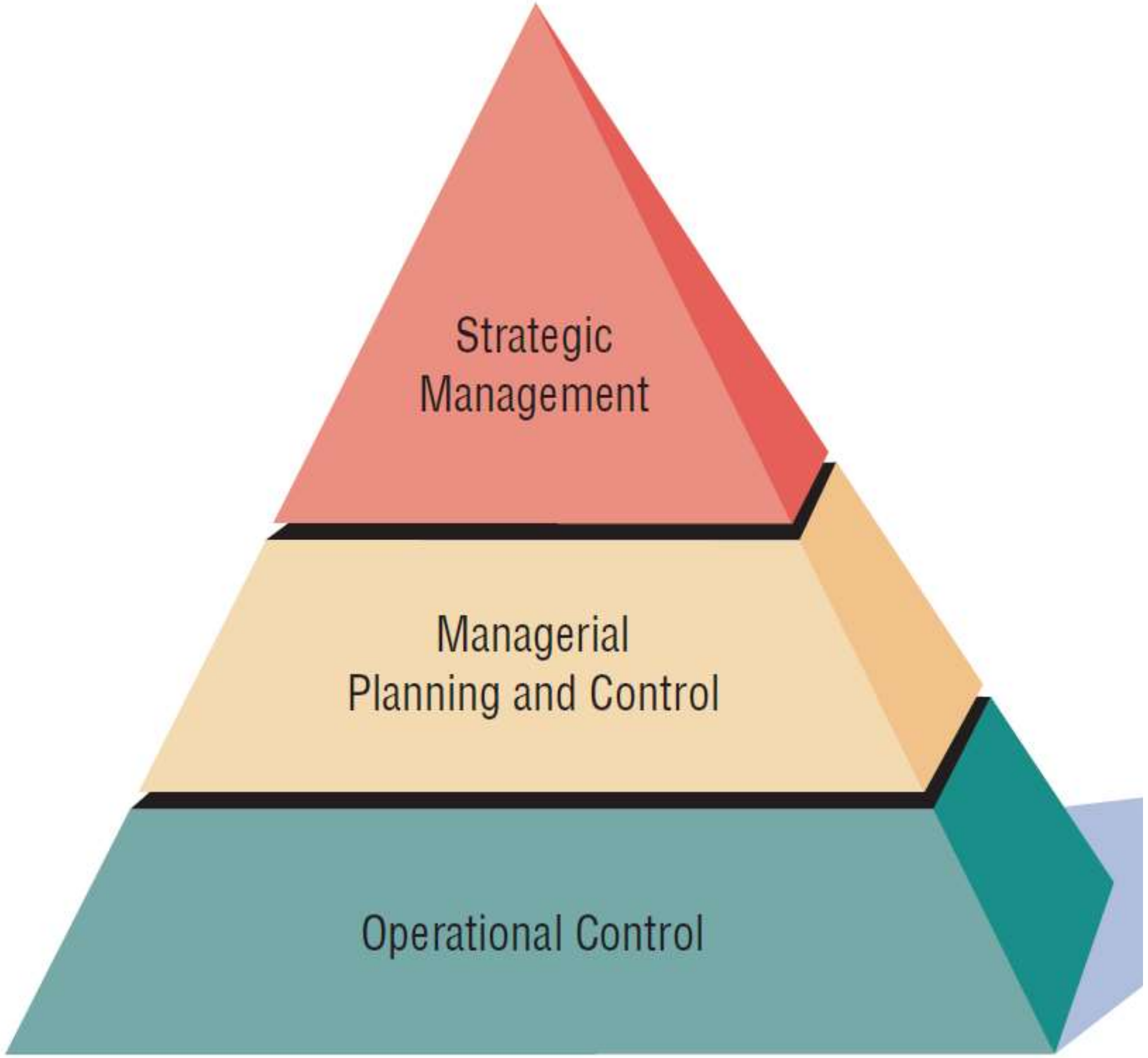
- As seen from Figure 2.18:
  - It easy for the analyst to identify processes
  - It also importantly aids in communication with other analysts on the team and with business executives
- The use case scenario is worthwhile:
  - Much information the users impart to the analyst already takes the form of stories
  - It is easy to capture the stories on a use case scenario form
- The use case scenario always documents the triggering event so that an analyst can trace the  steps that led to other use cases
- Because the steps performed are noted it is possible to employ use case scenarios to write logical processes

# Why Use Case Diagrams Are Helpful

- Use case diagrams are becoming popular because of their simplicity and lack of technical  detail
- They are used to show:
  - The scope of a system
  - The major features of the system
  - The actors who work with those major features
- The users can:
  - See the system
  - React  to it and provide feedback
  - They also may help to determine whether to build or buy the software

# Levels of Management

- Management in organizations exists on three broad horizontal levels:
  - operational control,
  - managerial planning and control (middle management), and
  - strategic management,
- as shown in Figure 2.19:
  - Each level carries its own responsibilities, and all work toward achieving organization goals and objectives in their own ways



# Managerial Levels

- In organizations there are a number of factors and considerations that feed into delivering effective management:
  - The organizational structure (there are different organizational structures)
  - Leadership style
  - Technological considerations
  - Organization culture
  - Human interaction
  - Socio-technological considerations
  - People (on all management levels)
- All these factors carry implications for the analysis and design of information systems

# Managerial Planning and Control

- Organizations must implement systems to realize:
  - Short term planning
  - Medium term planning
  - Long term planning
- These terms relate to the three levels of management
- Decision making (on all levels) may be clearly defined when viewed against the three levels of management but:
  - In the ‘real-world’ decisions may not be so clearly defined
  - Decisions may be partly operational and partly strategic

# Managerial Functions

- Operational control:
  - Forms the bottom tier of three-tiered management
  - Operations managers  make decisions using predetermined rules that have predictable outcomes when implemented  correctly
  - They make decisions that affect implementation in the areas of:
    - Work scheduling
    - Inventory control
    - Shipping (deliveries to customers)
    - Receiving (materials delivered to the organization)
    - Control of processes such as production
- Operations managers oversee the operating details of the organization

# Managerial Functions

- Middle management forms the second (or intermediate tier) of the three-tiered management  system
- . Middle managers make:
  - Short-term planning and control decisions about how resources may best be allocated to meet organizational objectives
  - Oversee the operations of the operations managers
  - In a hierarchical structure middle managers will be the link between the strategic management function and the operational management

# Managerial Functions

- Strategic management is the third level of three-tiered management control
- Strategic managers:
    - Consider the organization in the future
    - Make decisions that will guide middle and operations managers in the months and years ahead
    - Work in a highly uncertain decision-making environment

# Managerial Functions

- There are  sharp contrasts among the decision makers on many dimensions – for. Example:
    - Strategic managers have multiple decision objectives
    - Whereas operations managers have fewer (often single) objectives
    - It is often  difficult for high-level managers to identify problems as they may be relatively remote from day-to-day operations
    - Operations managers are close to day-to-day operations and can see problems more easily
    - Decisions may be partly operational and partly strategic
- Strategic management:
    - Must consider all stakeholders in the organization:
    - Many decisions on strategy going forward will result in difficult tough actions

# Managerial Functions

- Strategic managers:
  - Are faced with semi-structured problems
  - Whereas lower-level managers deal mostly with structured problems
  - The alternative solutions to a problem facing strategic managers are often difficult to articulate
  - The available alternatives that operations managers work with are usually easy to enumerate
  - Strategic managers most often make one-time decisions
  - Whereas the decisions made by operations  managers tend to be repetitive

# Implications for Information Systems Development

# Implications for Information Systems Development

- Each of the three management levels has different implications for developing information systems.
- Some of the information requirements for managers are clear cut whereas others are fuzzy and overlapping.
- Operations managers:
  - Need internal data and information that is often low level  and repetitive
  - Are driven by ‘real-world’ day-to-day data  and information that captures current performance
  - Are frequent users of online, real-time information resources
- Operations managers:
  - Have less need for information relating to past performance and the use of periodic information is only limited
  - Have little use for  external information that enables future projections
  - Make decisions using predetermined rules that have predictable outcomes

# Implications for Information Systems Development

- On the next management level use short- and longer-term information
- Middle managers ore often troubleshooting which needs ‘real-time’ data and information
- To ensure proper control:
    - They also need current  performance information based on a set of pre-defined standards and goals
- Middle managers are highly  dependent on internal information:
    - In contrast to operations managers
    - They have a high need for  historical information (and)
    - Information that allows for the prediction of future events and simulation of numerous possible scenarios

# Implications for Information Systems Development

- Strategic managers are different from both middle and operations managers in their information requirements
  - Strategic management is highly dependent on information from external sources that supply news of market trends and the strategies of competing corporations
  - Strategic management defines the organization as a whole
- Because the task of strategic management demands projections into the uncertain future:
  - Strategic managers have a high need for information of a predictive nature and information that allows creation of many different “what if” scenarios
  - They also need accurate assessments of:
    - Risk (including those posed to security of information system)
    - Strategic managers also exhibit strong needs for periodically reported information as they seek to adapt to fast-moving changes

# Collaborative Design

# Collaborative Design

- Collaborative design has many meanings in many fields including:
  - Architecture
  - Engineering
  - Medical systems
  - Computer science (which includes data science and information systems)
- In systems analysis and design:
  - Collaborative design means that stakeholders who are external (outside clients) along with those internal to the company follow  processes to share in designing a system that meets their goals
  - Internally: collaborators come from different management levels:
    - Strategic (top tier) / managerial (intermediate tier) / operations (bottom tier)
    - From different departments of the same organizational level

# Collaborative Design

- Research has identified that many internal design collaborations:
  - Rely on power relationships and information flows which could be based in part on hierarchical relationships in the organization
  - Information systems projects:
    - May go more smoothly if those in lower ranks of the organization (e.g., graphics designers) are given the opportunity to use their expertise to create initial designs that are then examined by those who are not directly in graphic design but are higher up in the IT department or organization
    - Giving power to those who possess technical or strategic expertise in preference to graphic  designers when using structured methods to begin a project may cause problems in the collaboration

# Collaborative Design

- For external collaborations it has been suggested that:
  - Attention be  paid to including (all) relevant stakeholders in appropriate information flows (and)
  - Emphasizing  the relationships that are formed between external and internal participants
- Recall INFO 102 (Introduction to Information Systems):
  - We introduced the research by Checkland & Holwell (1997):
    - Socio-technical information systems design
    - The soft systems methodology (SSM)
  - The observations made in the course textbook have a clear correlation to the observations made relating to collaborative design

# Organizational Culture

# Organizational Culture

- Organizational culture is an established area of research that has grown remarkably in the past decades a large body of documented research
- Just as it is appropriate to think of organizations as including many technologies – it is similarly appropriate to see them as hosts to multiple (often competing) subcultures
- There is little agreement on what precisely constitutes an organizational subculture:
  - It is agreed that competing subcultures may be in conflict (and)
  - Try to gain adherents to their vision of what the organization should be
- Ongoing research is attempting to identify the effects of virtual organizations and virtual teams on the creation of subcultures when members do not share a physical workspace but share tasks (refer to socio-technical IS research and the SSM introduced in INFO 102)

# Organizational Culture

- Organizations have cultures and subcultures
- Rather than think about culture as a whole – it is more useful to think about the researchable  determinants of subcultures such as shared verbal and nonverbal symbolism
- Learn from verbal and nonverbal symbolism
- Verbal symbolism:
  - Myths / metaphors / visions / humour
- Non-verbal symbolism:
  - Shared artefacts:
    - Trophies / rites / rituals
  - Promotions
  - Birthdays, etc.
  - Clothing worn
  - Office placement and decorations

# Organizational Culture

- Verbal symbolism includes shared language used to:
  - construct, convey, and preserve subcultural myths, metaphors, visions, and humour
- Nonverbal symbolism includes:
  - Shared artefacts, rites, and ceremonies; clothing of decision makers and workers; the use, placement, and decoration of offices; and rituals for celebrating members' birthdays, promotions, and retirements. Subcultures coexist within "official" organizational cultures.
- The officially sanctioned culture may prescribe:
  - Dress code / suitable ways to address superiors and co-workers / proper ways to deal with the outside public

# Organizational Culture

- Subcultures may be powerful determinants of information requirements, availability, and use.
- Organizational members may belong to one or more subcultures in the organization.
- Subcultures may exert a powerful influence on member behaviour, including sanctions for or against the use of information systems
- Understanding and recognizing predominant organizational subcultures may help a systems analyst overcome the resistance to change that arises when a new information system is installed.
    - For example, the analyst might devise user training to address specific concerns of organizational subcultures

# Technology’s Impact on Culture and Slack

- Technology is changing the culture of organizations and teams
- Slack is an employer-sanctioned  social media platform, or workplace-messaging app.
    - Conversations between co-workers can be  public, private, or something in between.
    - Conversations can be put together as a group or as a  person-to-person conversation.
    - Channels can be devoted to seemingly trivial concerns such as  popular snacks or employee cafeteria food, but its main purpose is to foster collaboration and communication among officer workers.
    - It is less formal that email and might be more popular  with those in the millennial age group.

# Slack

- Slack can shape a culture and it can become an office culture complete with:
  - Programmable pranks
  - Featuring less formal interactions that email
  - Helping team members to avoid the difficulty of expressing themselves face-to-face in a meeting (or even one to one)
- Slack has both public and private channels:
  - Public channels are open to all Slack team members with messages posted to them archived (persistent storage) and searchable by the entire work team
  - Private channels are circumscribed so that they are available to a limited number of team members and are joined by invitation
  - You must be a private channel member to be able to view and search the contents of the private channel.
- Slack also has direct messages (DM) or group messages (group DMs) that are used for rapid, private messaging between two or more members of a work team
  - The searchability depends on whether you are a recipient of the DM
  - In which case you can view and search the contents.
- Slack and many other technology-based tools can be very helpful in team communication.

# Chapter  #2 Summary

# Summary

- There are three broad organizational fundamentals to consider when analysing and designing information systems:
  - The concept of organizations as systems
  - The various levels of management
  - The overall  organizational culture
- Organizations are complex systems composed of interrelated and interdependent subsystems.
- Moreover, systems and subsystems are characterized by their internal environments on a continuum from open to closed:
  - An open system allows free passage of resources (people, information, materials) through its boundaries
  - Closed systems do not permit free flow of input or output

# Summary

- Organizations and teams also can be organized virtually:
  - With remote members who are not in the same physical workspace connected electronically.
- Enterprise resource planning systems are integrated organization (enterprise) information systems developed with customized, proprietary software that help the flow of information between the functional areas in the organization.
- There are many ways to graphically depict the system.
  - An analyst should Choose among these tools early on to get an overview of the system.
  - These approaches include:
    - Drawing context-level data flow diagrams
    - Capturing relationships early on with E-R diagrams
    - Drawing use case diagrams (and/or) writing use case scenarios based on user stories.

# Summary

- Using these diagrams and techniques at the beginning of an analysis can help in:
  - Defining the boundaries and scope of the system
  - Can help to bring into focus:
    - Which people and systems are external to the system being developed
  - Entity-Relationship diagrams help a systems analyst to:
    - Understand the entities and relationships that comprise an organizational system
  - Entity-Relationship diagrams can show many relationships:
    - One-to-one relationships
    - One-to-many relationships
    - Many-to-one relationships
    - Many-to-many relationships

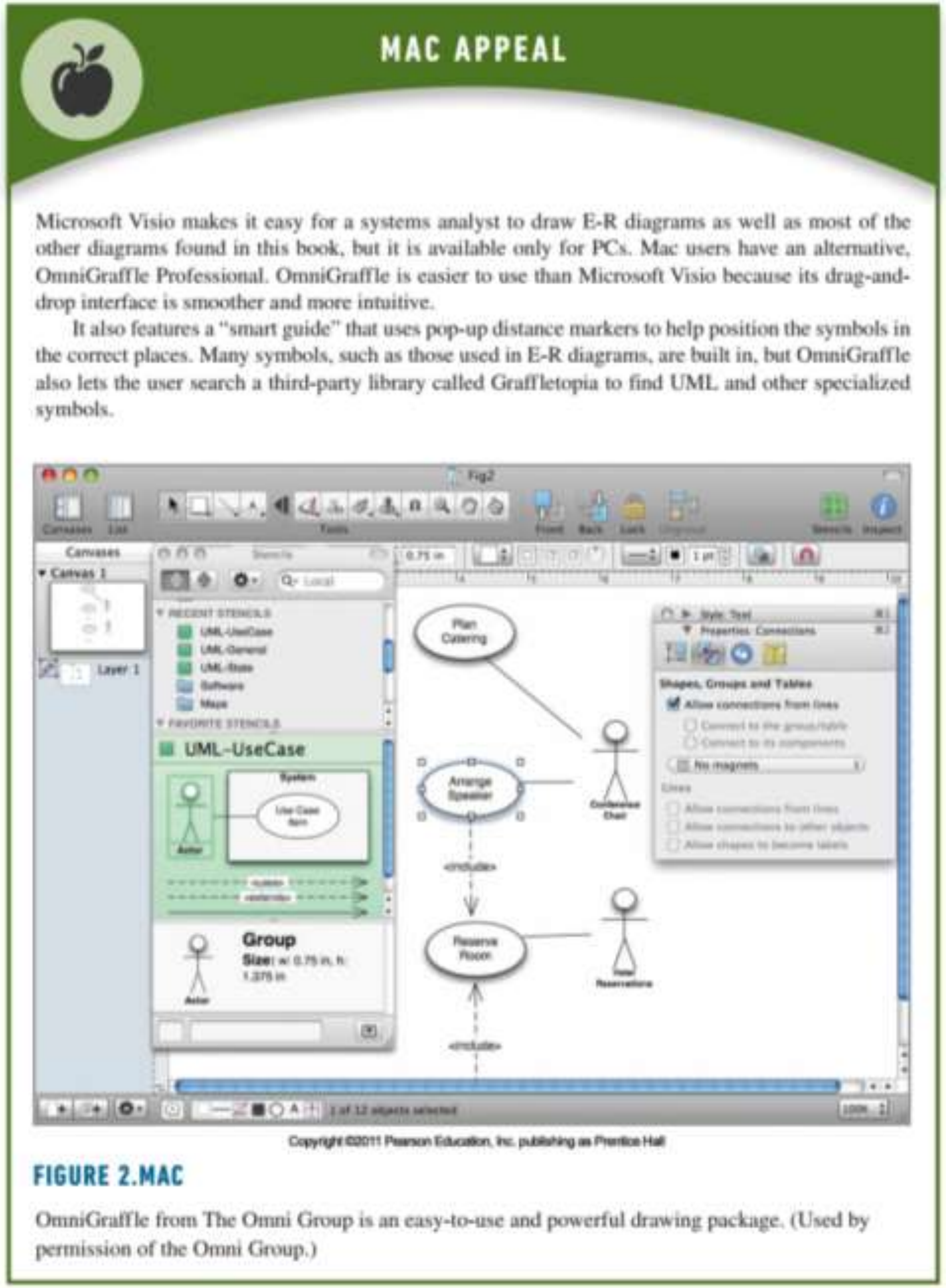
# Summary

- The three levels of managerial control are:
    - Operational
    - Middle management
    - Strategic management
- The time  horizon of decision making is different for each level
- Organizational cultures and subcultures are important determinants of how people use information  and information systems
    - By grounding information systems in the context of the organization as a larger  system
    - It is possible to realize that numerous factors are important and should be taken into account when  ascertaining information requirements and designing and implementing information systems
- Technologies such as the work-based social platform Slack can be used to create or reinforce elements of the organization's  culture or subculture

# Case Study

# Mac Appeal

- The Figure 2.MAC shows:
  - The MacBook Pro GUI
  - A tool to develop UML diagrams
  - A simple Use-Case diagram
- There are a range of both free and purchased software tools including:
  - Microsoft Visio (expensive)
  - Free apps for both Windows and Apple OS
- Remember:
  - Drawing ‘pretty’ models is easy
  - But ‘pretty’ models may not be correct of logical
- Careful data collection (requirements) and design is essential



MAC APPEAL

Microsoft Visio makes it easy for a systems analyst to draw E-R diagrams as well as most of the other diagrams found in this book, but it is available only for PCs. Mac users have an alternative, OmniGraffle Professional. OmniGraffle is easier to use than Microsoft Visio because its drag-and-drop interface is smoother and more intuitive.

It also features a “smart guide” that uses pop-up distance markers to help position the symbols in the correct places. Many symbols, such as those used in E-R diagrams, are built in, but OmniGraffle also lets the user search a third-party library called Graffletopia to find UML and other specialized symbols.

Copyright ©2011 Pearson Education, Inc. publishing as Prentice Hall

**FIGURE 2.MAC**

OmniGraffle from The Omni Group is an easy-to-use and powerful drawing package. (Used by permission of the Omni Group.)