MIS
(Management Information System)

Sharif University of Technology

Session # 7

Session schedule

- *Contents*
  - *Systems Analysis and Design*
    - *Planning the approach*
    - *Asking questions and collecting data*
    - *Recording the information*
    - *Interpreting the information collected*
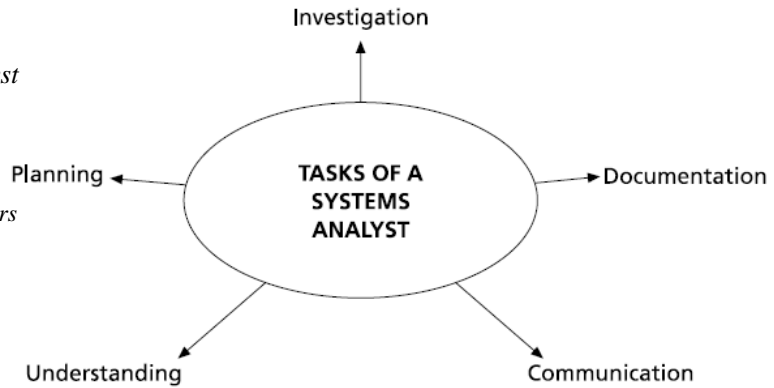   - *Specifying the requirement*

Information system development

- *System Analysis*
  - *Five areas of a system analyst tasks:*
    - *Investigation*
    - *Communication with customers*
    - *Documentation*
    - *Understanding*
    - *Preparation & planning*

Information system development

- *System Analysis*
  - *System analysis process:*
    - *The PARIS Model*
    Analysis can be considered to be a Five-stage process
    - *Planning the approach*
    - *Asking questions and collecting data*
    - *Recording the information*
    - *Interpreting the information collected*
    - *Specifying the requirement*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Typically, the analyst collects a considerable amount of information during the investigation phase, which may include*
      - *Interview reports,*
      - *Observation records,*
      - *Sample documents,*
      - *Completed questionnaires and*
      - *Lists of problems and requirements.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Structured analysis and design views information systems principally in two ways:*
      - *Data – the information that the system records*
      - *Processing – what the system does with this data.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Object-oriented (OO) methods*

    - *Object-oriented (OO) methods allow the developer to exploit the technology of distributed computing environments, Internet-based systems and communications software and tools.*

    - *The OO approach originated in software engineering, and it is commonplace to find coded examples of objects in texts that describe OO from the software engineer's perspective.*

    - *The term business process re-engineering is often applied to this type of development, and the experienced system analyst will approach such projects with caution.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Principles of OO*

    Three terms most frequently cited are:
      - *Inheritance,*

      - *Encapsulation and*

      - *Polymorphism*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Principles of OO-Inheritance*
      - *Inheritance is derived from the idea of objects forming classes.*

      - *In simple terms, an object can be defined at a high or superclass level with certain characteristics and certain procedures that are then inherited as properties by the lower or subclasses of the object.*

      - *Objects and object classes are fundamental building blocks in the OO approach.*

      - *Identifying classes is both an analysis and a design activity, and the designer must make decisions about the level of generalization to be applied to the system.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Principles of OO-Inheritance*
      - *The definition of an entity in structured systems is often given as*
        'something the system wishes to hold data about'.

      - *A formal definition of an object is*
        'An abstraction of something in a problem domain, reflecting the capabilities of the system to keep information about it, interact with it, or both'.

      - *The analyst must identify abstractions in the objects to be able to define the super classes.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Principles of OO-Encapsulation*
      - *In OO the development goal is set so data the data held by the object can be accessed, read or updated only by the operations that are defined for that object.*
      - *The object-oriented method requires software to pass the request from one object to another object so that the operations can be performed by each object in their own way.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Principles of OO-Polymorphism*
      - *The term polymorphism first occurred in the English language in 1839, and was defined as 'the occurrence of something in several different forms'.*
      - *It has been adopted in OO to encourage designers to use common operations to do similar things.*
      - *Polymorphism encourages the designer to identify similarities between the ways objects perform and to define them in abstract terms.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Object-Oriented Models*
      - *RUP is made up of three model types:*
        - *Business system models   --- Use Case Diagrams*

        - *Static structure models   --- Class Inheritance Diagrams*

        - *Dynamic behavior models --- State Transition Diagrams*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Use Case Diagrams*
    - *A use case diagram is constructed to show the required functionality of the system in the analysis phase and to specify the actual behavior of the system in the design phase.*

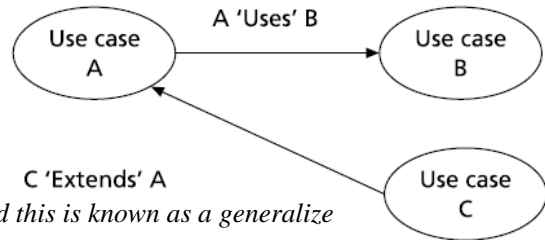   - *The use case is said to be associated with an actor, often represented by a matchstick man.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Use Case Diagrams*
    - *A relationship can exist between use cases, and this is known as a generalize relationship.*

    - *A use case may use or extend another use case. When a function is required by a number of use cases, for example a function to locate a specific instance of an object, then each of the use cases can be said to use it.*

    - *A use case can extend another use case when it is invoked by the first use case to provide some additional functionality.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Use Case Diagrams*
    - *The use case diagram is a business model for an OO specification. It can used for analysis purposes, to record system requirements and enable the analyst to understand how the current system works, or for design purposes, to show he system behavior and the meaning of a particular set of procedures.*
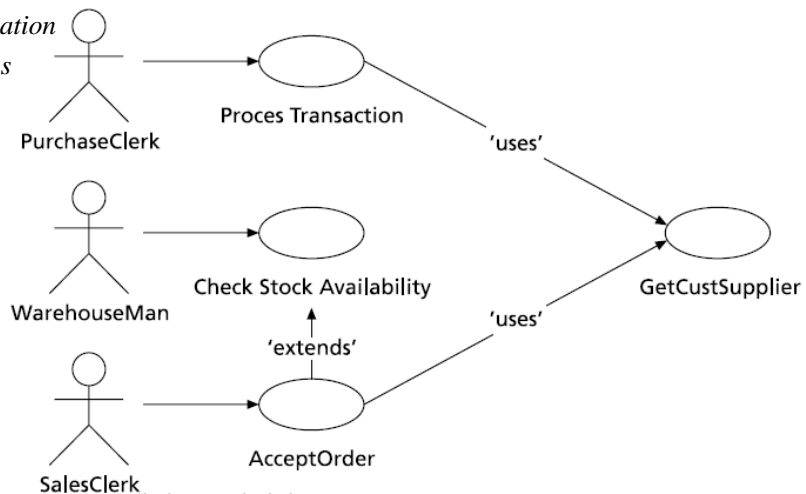
    - *In some circumstances it is preferable to use a more detailed breakdown of the actions performed by the user or actor and the responses from the system.*

Information system development

▪ *System Analysis*
  ▪ *Recording the information*
    ▪ *Use Case Diagrams*

Information system development

▪ *System Analysis*
  ▪ *Recording the information*
    ▪ *Class Inheritance Diagrams*
    ▪ *Class inheritance diagrams are the static structure models of OO developments.*

    ▪ *Objects are said to belong to classes. Objects can have attributes and can exist in different states.*

    ▪ *Objects are often similar to entities in structured systems analysis, although an object is an abstraction of anything within the domain of the system.*

    ▪ *Objects are an abstraction of a thing found in the real world, and they are said to have a state, attributes and behaviour.*
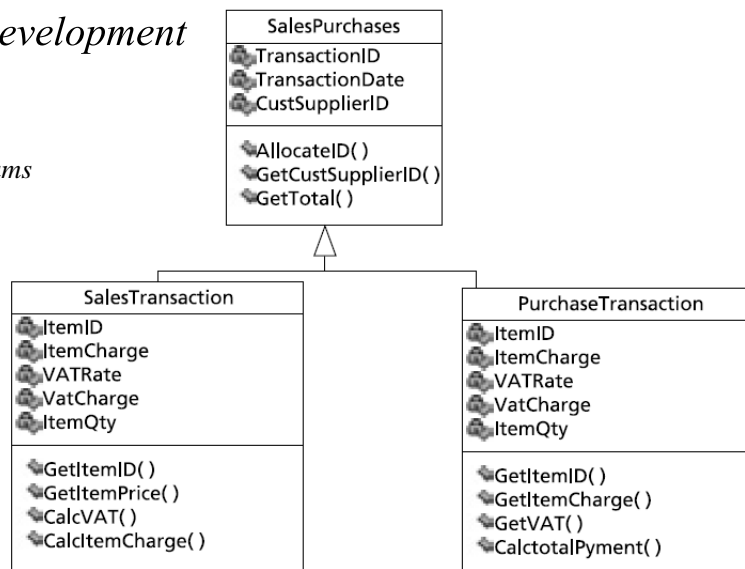
Information system development

- *System Analysis*
  - *Recording the information*
    - *Class Inheritance Diagrams*
    - *Identifying objects is often the first stage in OO analysis, and initially the analyst can safely include anything as an object that seems relevant to the system.*

    - *Objects are fitted into classes by establishing their similarities. A class defines a set of objects; an occurrence of a single object is known as an instance of a class.*

      e.g.
    - *The analyst would notice the similarities between sales invoices issued by a company and the purchase invoices that it receives. This might lead to the definition of the abstract class 'transaction' which has attributes and methods common to both purchase and sales invoices.*
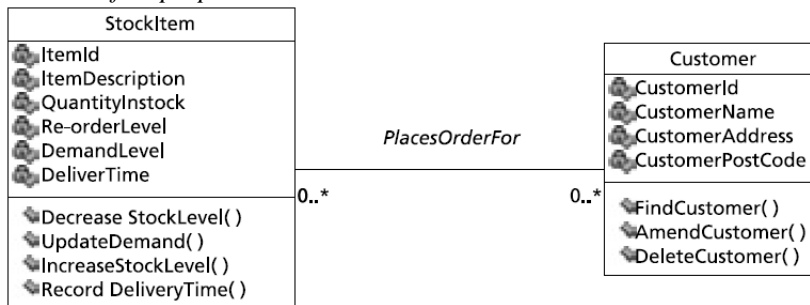
Sharif University of Technology
MIS (Management Information System), Session # 7   *19*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Class Inheritance Diagrams*

SalesPurchases
- TransactionID
- TransactionDate
- CustSupplierID

- AllocateID( )
- GetCustSupplierID( )
- GetTotal( )

SalesTransaction
- ItemID
- ItemCharge
- VATRate
- VatCharge
- ItemQty

- GetItemID( )
- GetItemPrice( )
- CalcVAT( )
- CalcItemCharge( )

PurchaseTransaction
- ItemID
- ItemCharge
- VATRate
- VatCharge
- ItemQty

- GetItemID( )
- GetItemCharge( )
- GetVAT( )
- CalctotalPyment( )

Sharif University of Technology
MIS (Management Information System), Session # 7   *20*

Information system development

▪ *System Analysis*
  ▪ *Recording the information*
    ▪ *Class Inheritance Diagrams*
    ▪ *Classes of object interact. These interactions are represented by a line connecting the two objects, named to signify the association. The association is given a meaningful name that is indicative of its purpose.*

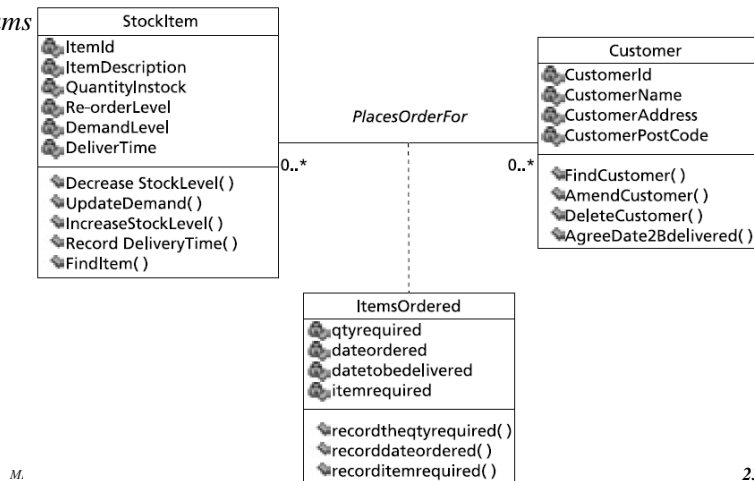## Information system development

▪ *System Analysis*
  ▪ *Recording the information*
    ▪ *Class Inheritance Diagrams*
    ▪ *An association can have attributes.*

    ▪ *In data modeling the many-to-many relationship is eliminated in favor of two one-to-many relationships linked to the 'intersection' data. In an OO model we use an 'association' class to perform the same service. Our model above now has the additional class of ItemsOrdered linked to the association between customer and item.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Class Inheritance Diagrams*

M.

23

Information system development

- *System Analysis*
  - *Recording the information*
    - *Class Inheritance Diagrams*
    - *Although relational databases dominate the market for accounting and administrative systems there are some business areas that they are not suited to, and the developer must look to OO databases to meet requirements.*
    - *Cases like simultaneous decision-making to take place and reduction the throughput time for the transaction needs OO concepts.*
    - *Integrating this type of application with applications using the formatted data structures found in a relational database system forces the analyst to apply OO principles alongside the normalization and entity modeling approach used by the data modeler.*
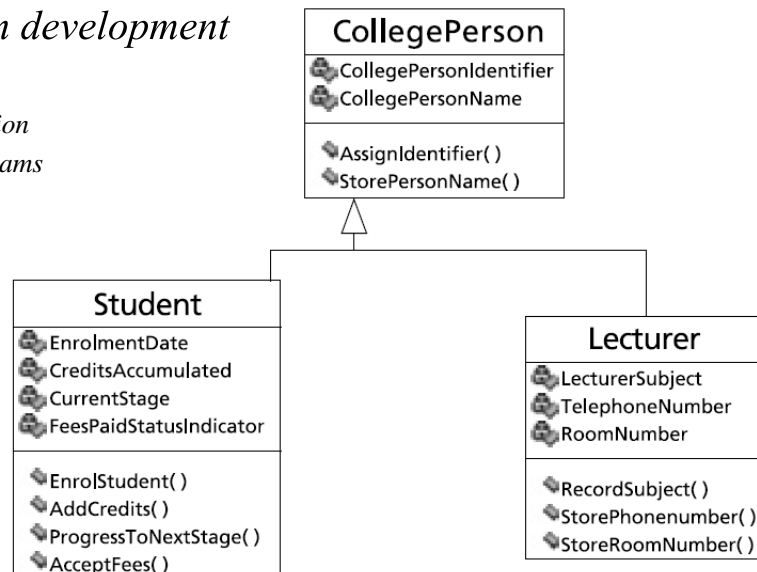
Information system development

- *System Analysis*
  - *Recording the information*
    - *State transition diagrams*
    - *Objects have a state. This is a finite, non-instantaneous period of time in which the object fulfills a condition.*
    - *It is the event that causes the object to move from one state to the next. Certain classes of object will change their state more frequently than others.*
    - *The state chart diagram has a start state represented by a solid circle and a final state represented by a solid circle with a surrounding ring. Each state change is annotated with the change event that causes it to happen.*

Information system development

- *System Analysis*
  - *Recording the information*
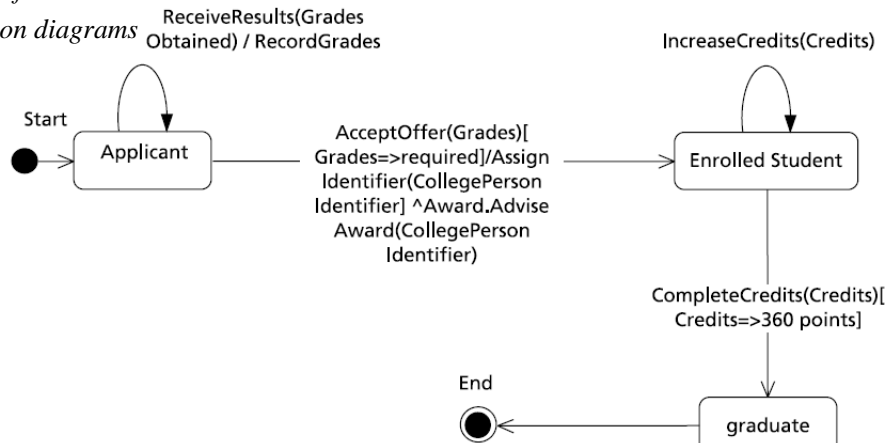   - *State transition diagrams*

CollegePerson
- CollegePersonIdentifier
- CollegePersonName
- AssignIdentifier( )
- StorePersonName( )

Student
- EnrolmentDate
- CreditsAccumulated
- CurrentStage
- FeesPaidStatusIndicator
- EnrolStudent( )
- AddCredits( )
- ProgressToNextStage( )
- AcceptFees( )

Lecturer
- LecturerSubject
- TelephoneNumber
- RoomNumber
- RecordSubject( )
- StorePhonenumber( )
- StoreRoomNumber( )

Information system development

- *System Analysis*
  - *Recording the information*
    - *State transition diagrams*

ReceiveResults(Grades Obtained) / RecordGrades

IncreaseCredits(Credits)

Start

Applicant

AcceptOffer(Grades)[Grades=>required]/Assign Identifier(CollegePerson Identifier] ^Award.Advise Award(CollegePerson Identifier)

Enrolled Student

CompleteCredits(Credits)[Credits=>360 points]

End

graduate

Information system development

- *System Analysis*
  - *Recording the information*
    - *State transition diagrams*
    - *Event naming can be done to a syntax, which has been defined for UML as*

```
event (arguments) [condition] / action ^ target.SendEvent
(arguments)
```
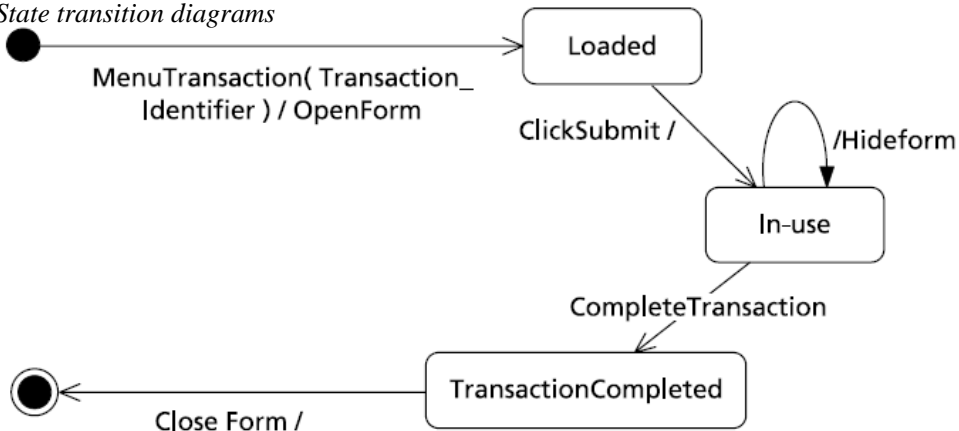
    - *The conditions must be satisfied for the transitions to take place.*
    - *The action, like the event, may have parameters with it.*
    - *The SendEvent denotes a type of event that sends a message to another object, and this too may be accompanied by parameters.*

Information system development

- *System Analysis*
  - *Recording the information*
    - *State transition diagrams*

Information system development
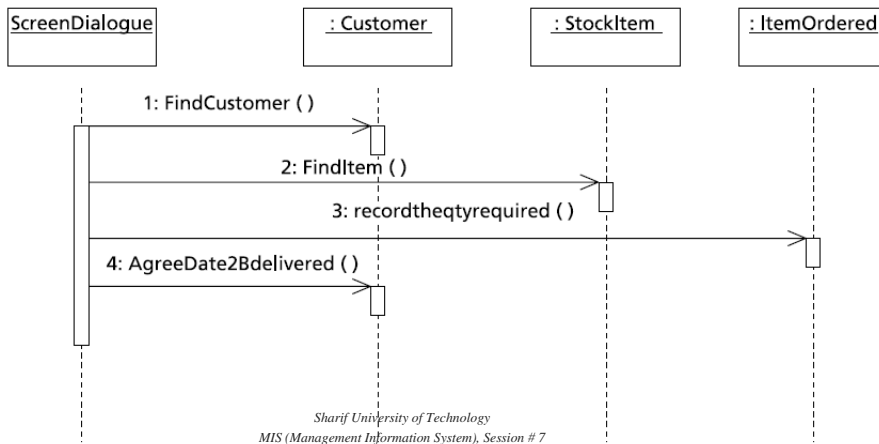
- *System Analysis*
  - *Recording the information*
    - *State transition diagrams → Sequence diagram*
    - *Objects interact to form complete systems.*

    - *The behavior of object interaction is shown with the sequence diagram, which shows the exchange of messages between different objects in a time sequence.*
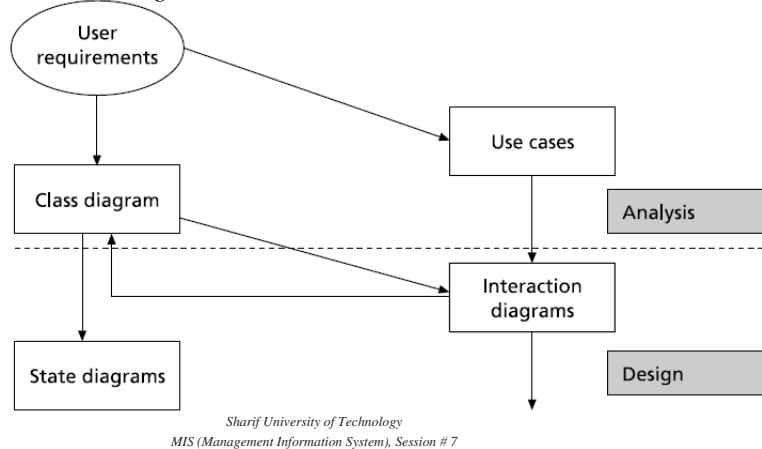
Information system development

- *System Analysis*
  - *Recording the information*
    - *State transition diagrams → Sequence diagram*

Information system development

- *System Analysis*
  - *Recording the information*
    - *Object Oriented modeling*