



SOFTWARE ENGINEERING

Week 6

Analysis Model

Agenda

1. Requirements Analysis
2. Structured Analysis
 1. Data Model: Database objects and relations
 2. Functional Model: Data flow
 3. Behavioural Model: Control flow, Events and states
3. Object Oriented Analysis

Analysis Model 2

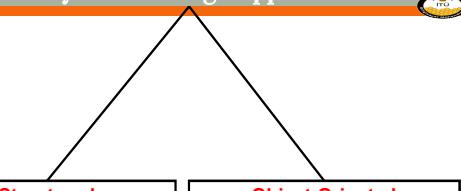
1. Requirements Analysis ←
2. Structured Analysis
 1. Data Model
 2. Functional Model
 3. Behavioural Model
3. Object Oriented Analysis

Requirements Analysis

80 6.1 03

Analysis

Analysis and Design Approaches



Structured Analysis and Design

Data / Control Flow Diagrams, (DFD / CFD)

Object-Oriented Analysis and Design

Unified Modeling Language Diagrams (UML)

Analysis Model 4

Elements of Analysis Model

- The Statement of Software Scope provides the basis for analysis modelling.
- The following models are built during analysis:
 1. Data model: Database objects and relations
 2. Functional model: Data flow
 3. Behavioural model: Control flow, Events and states

Analysis Model 5

Modeling the Data Domain

- 80 Define data objects
- 80 Establish data relationships
- 80 Specify data content

Analysis Model 6

Modelling the Functions

Basic Idea:

- Software transforms data
- To achieve this it must perform at least three generic functions: **input, processing, output**
- Identify functions that transform data objects

Begin with a context level diagram (**level 0**)

Continue with more functional details in refined levels until all system functionality is represented

Analysis Model 7

Modeling the Behaviour

Basic Idea:

- Most software responds to **events** from the outside world
- This characteristic forms the basis of the behavioral model
- A computer program always exists in some state: an externally observable mode of behaviour (e.g. waiting, computing, printing, polling) that is changed only when some event occurs

Indicate different **states** of the system

Specify events that cause the system to change state

Analysis Model 8

- Requirements Analysis
- Structured Analysis 

 - Data Model
 - Functional Model
 - Behavioural Model

- Object Oriented Analysis

Structural Analysis

6.2.03

Analysis

- Requirements Analysis
- Structured Analysis 

 - Data Model
 - Functional Model
 - Behavioural Model

- Object Oriented Analysis

The Data Model

6.2.103

Analysis

The Data Model

Data modelling is also called Database Modelling.

In data modelling, **Entity-Relationship Diagrams** are used.

Also a data dictionary is defined for important data items.

Entity Symbols (Bachman notation)



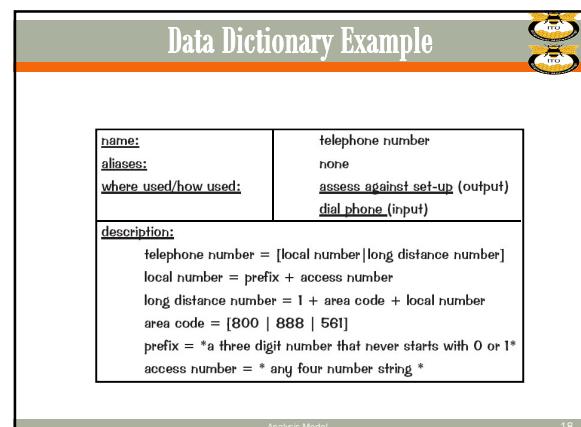
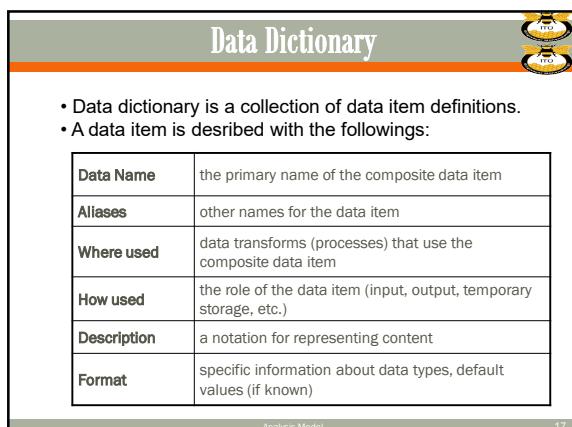
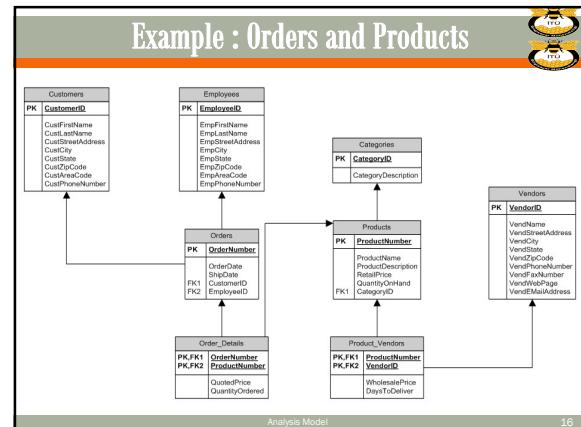
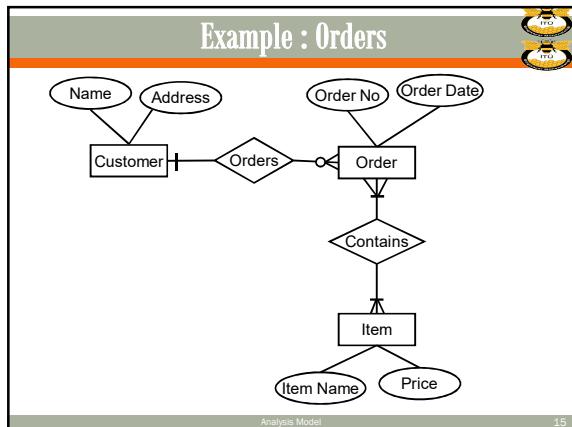
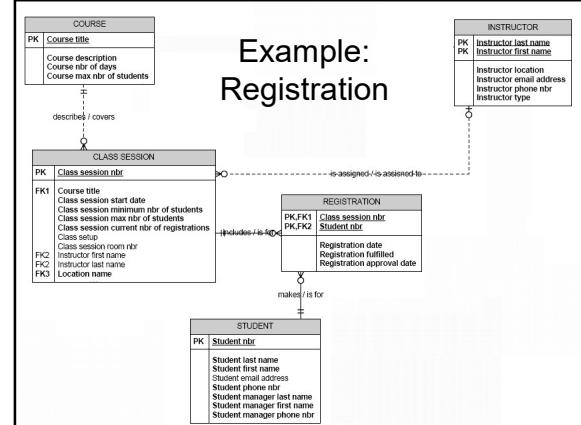
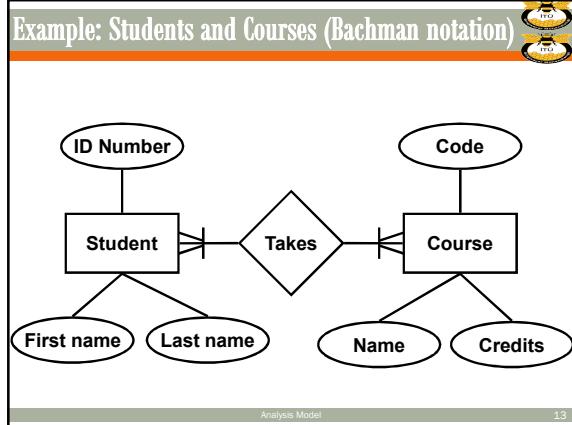
Analysis Model 11

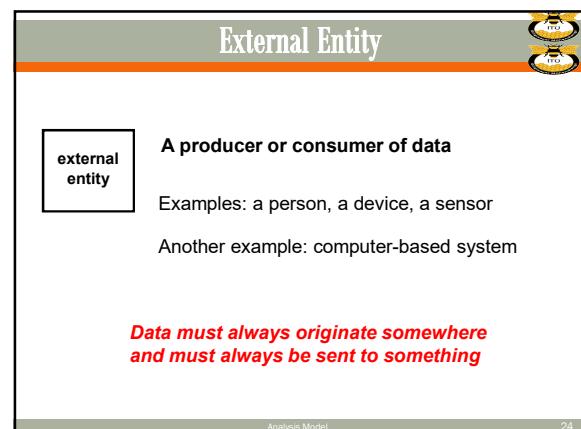
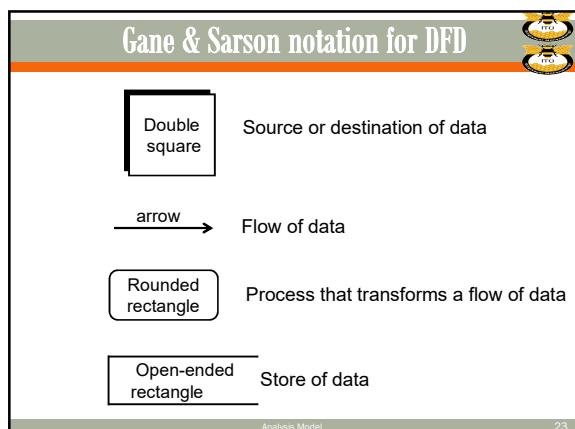
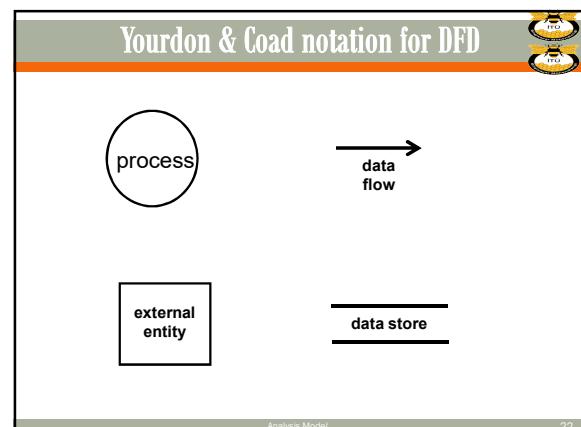
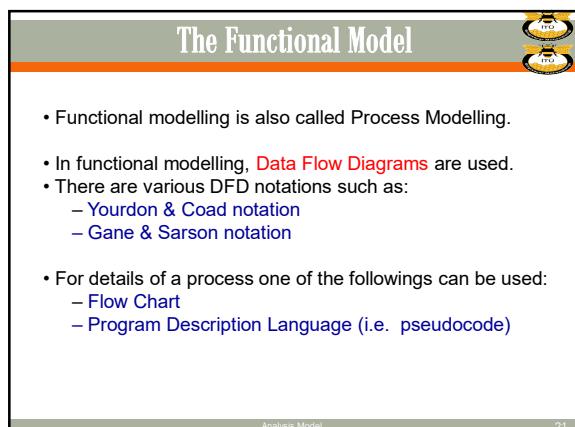
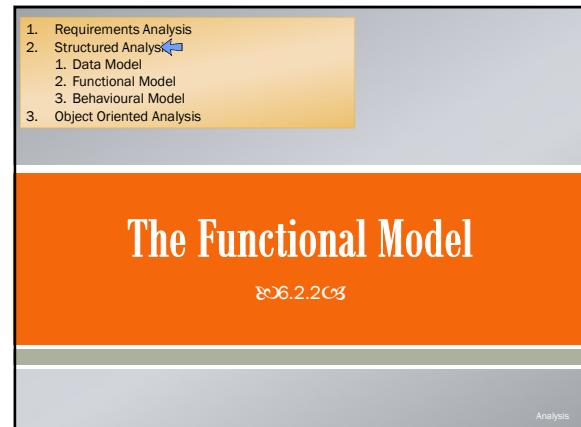
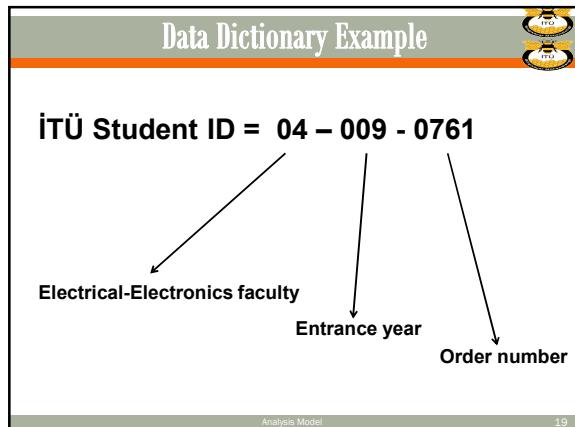
Relationship Symbols

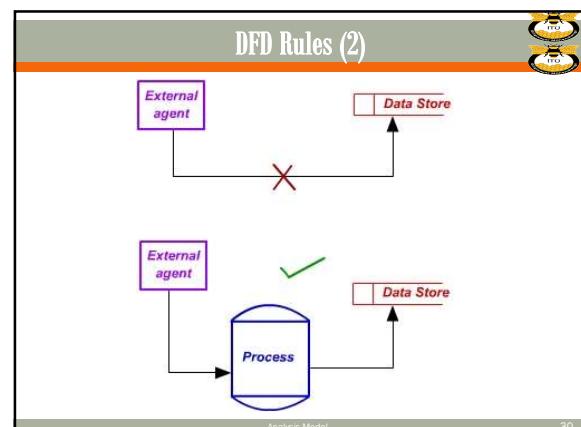
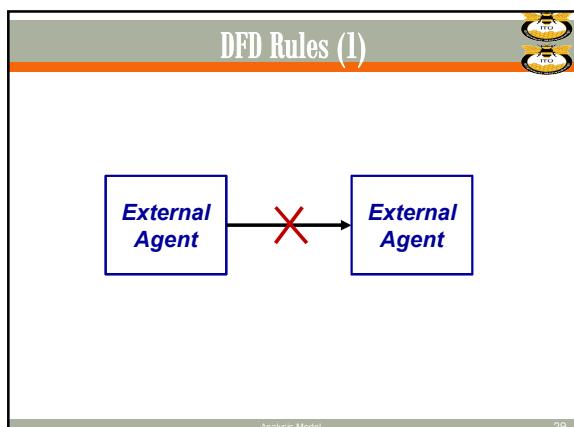
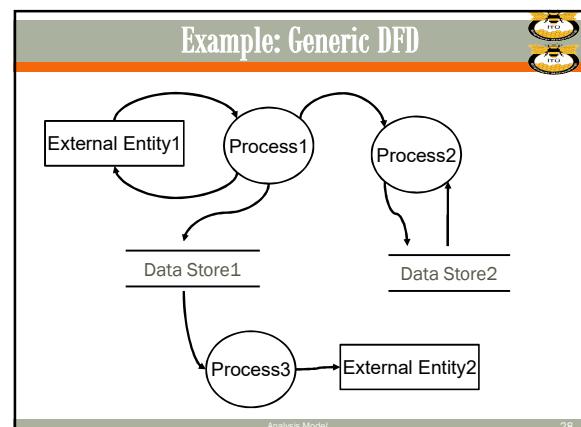
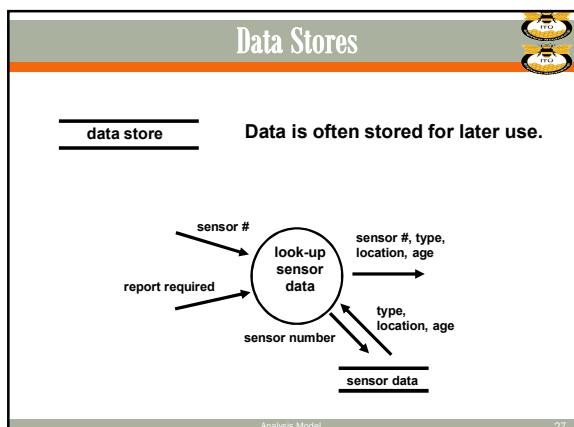
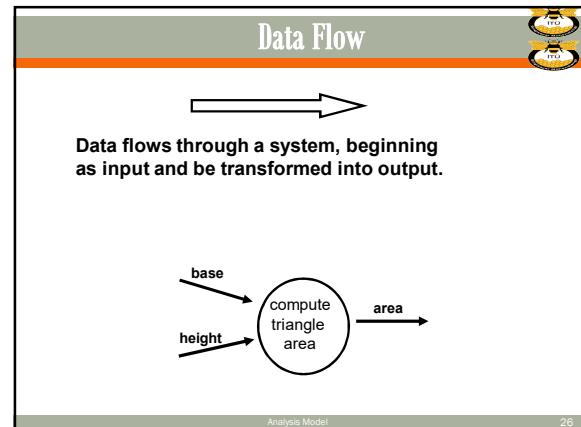
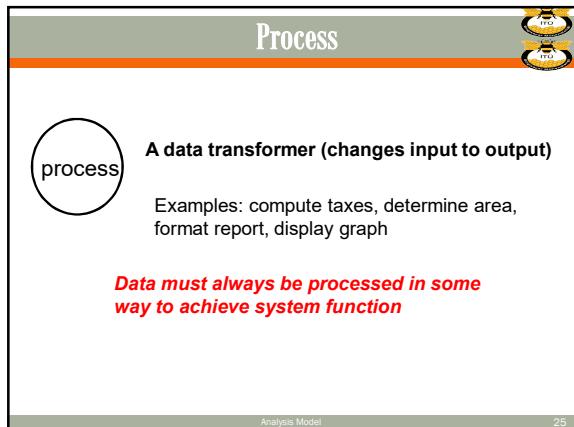
	One to one, mandatory
	One to one, optional
	One to many, mandatory
	One to many, optional

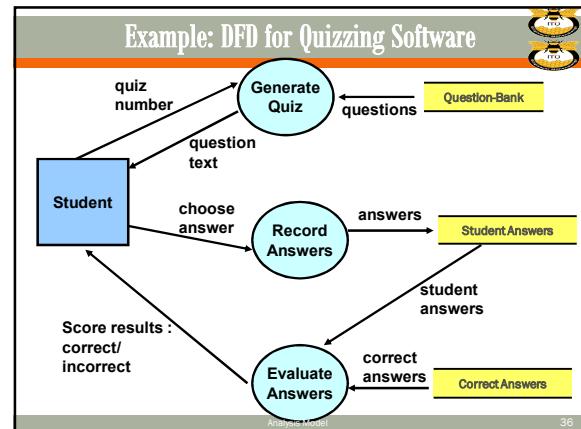
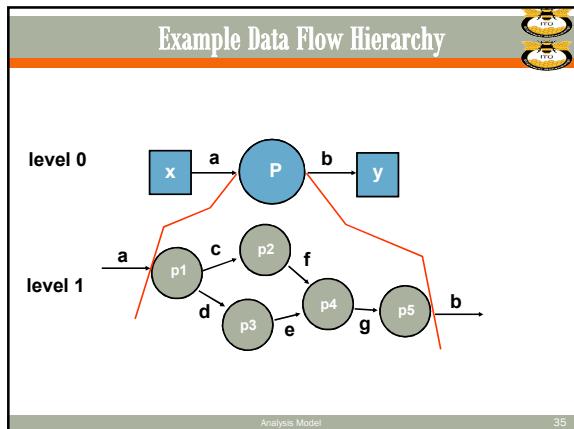
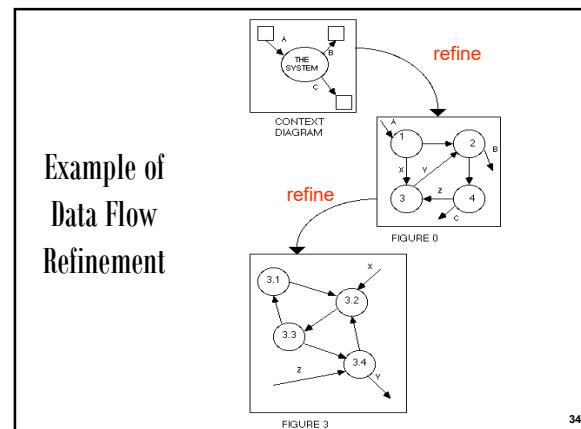
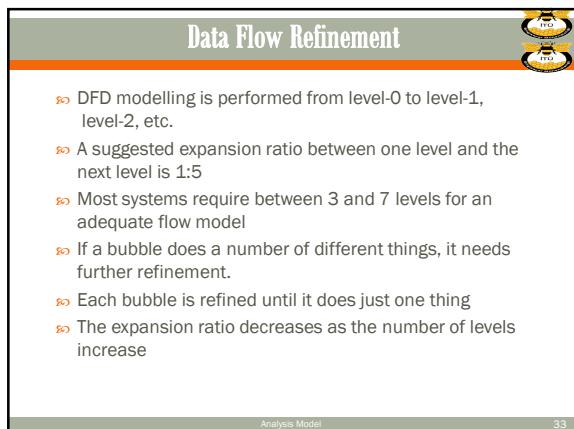
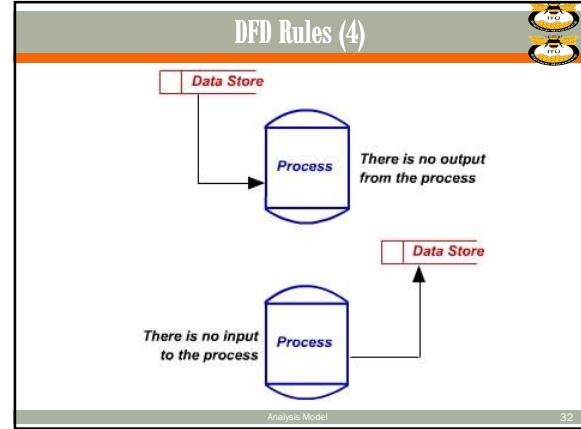
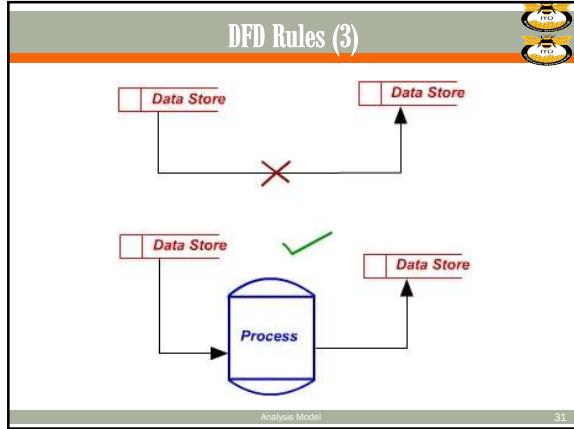
Cardinality **Modality**

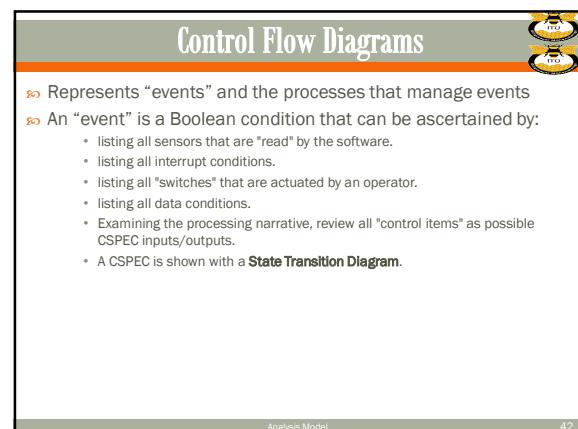
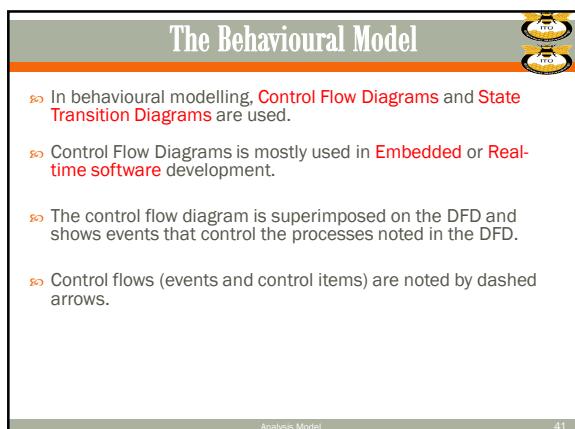
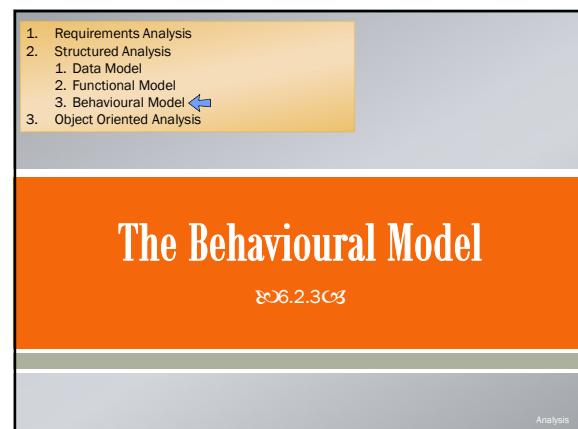
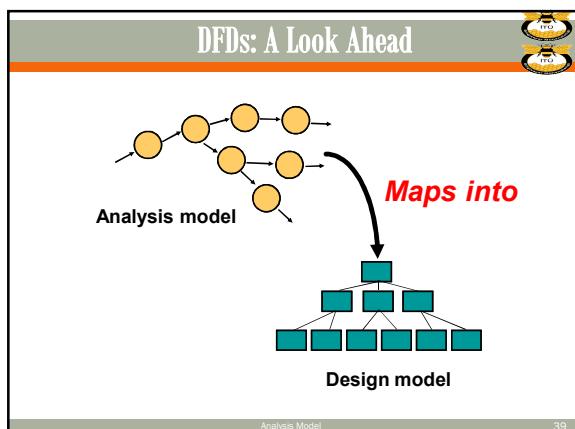
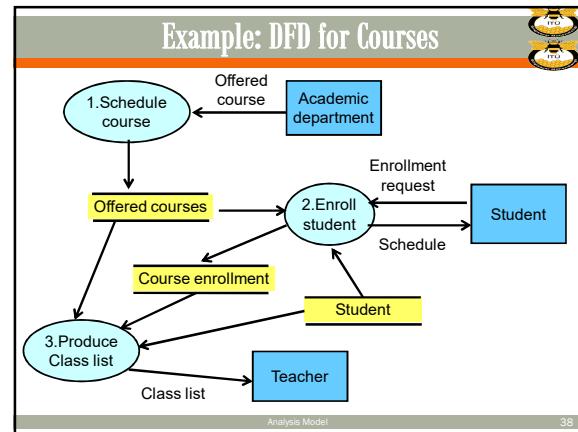
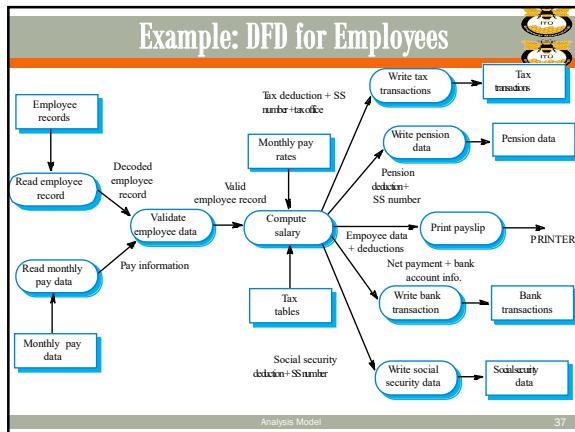
Analysis Model 12

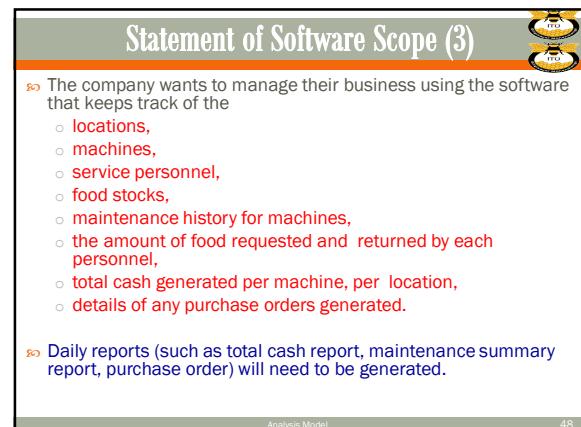
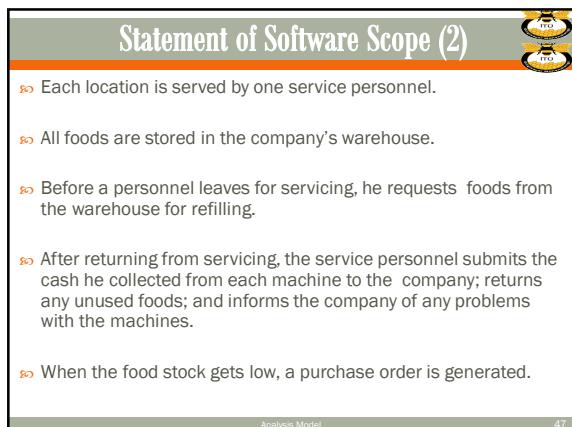
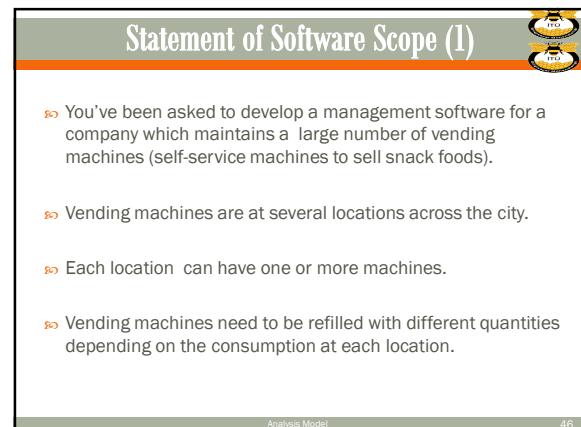
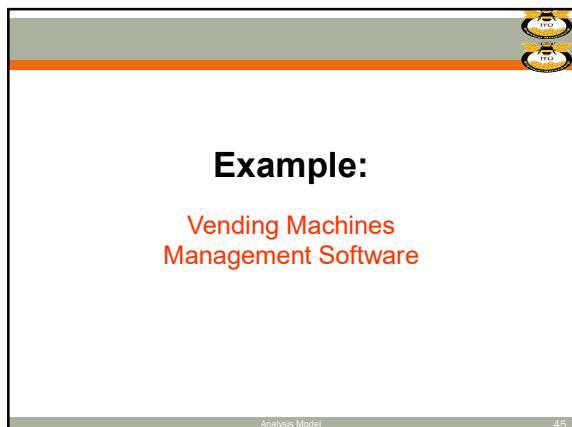
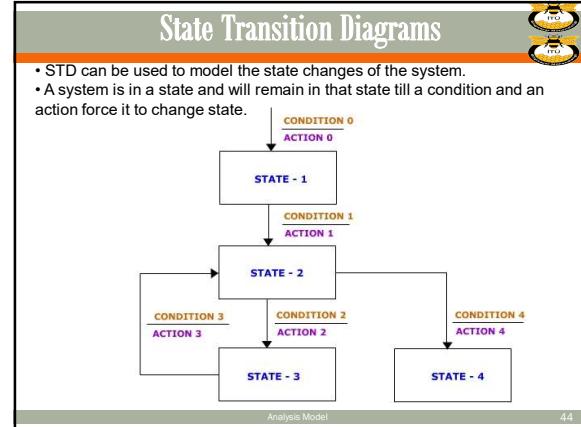
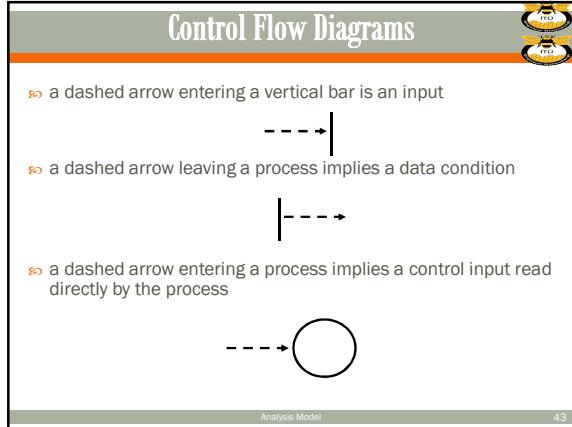


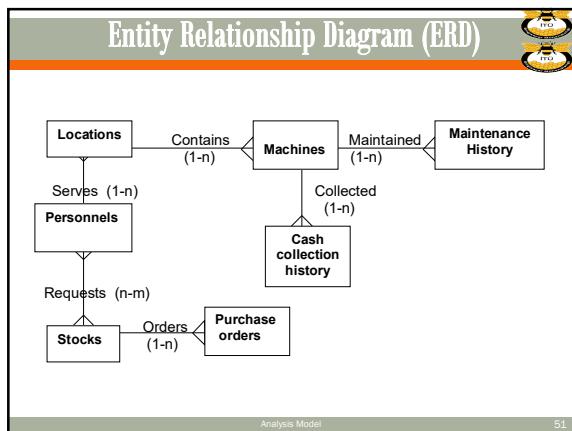
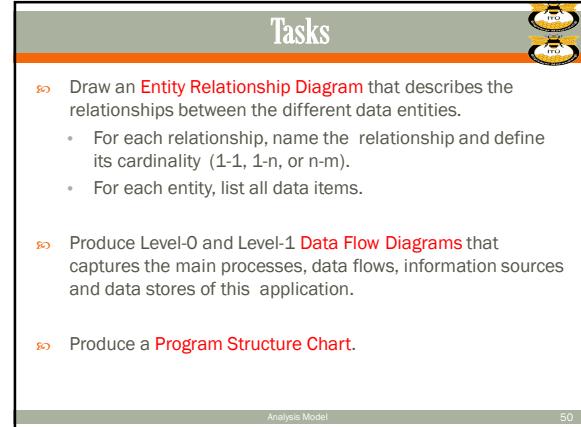
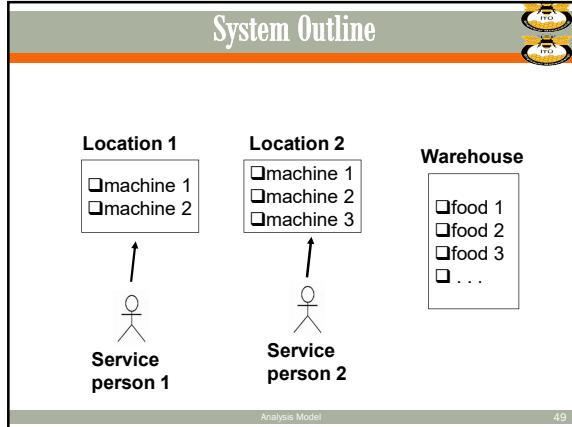








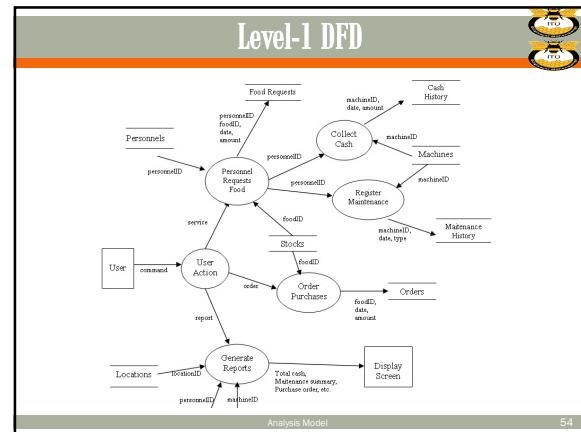
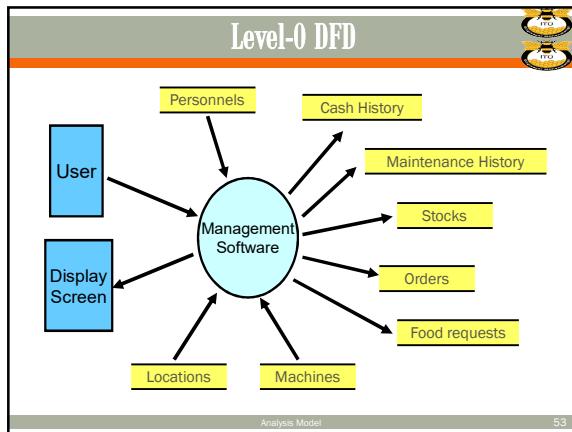


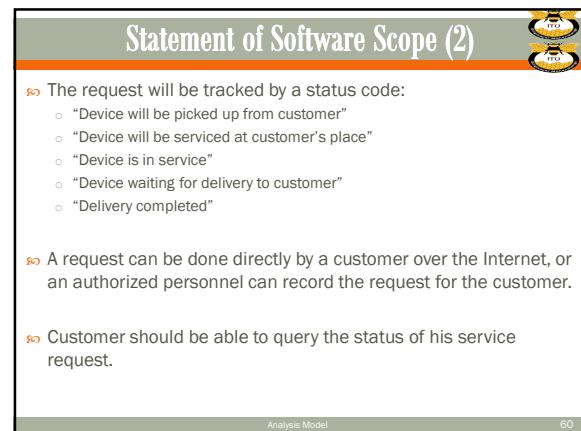
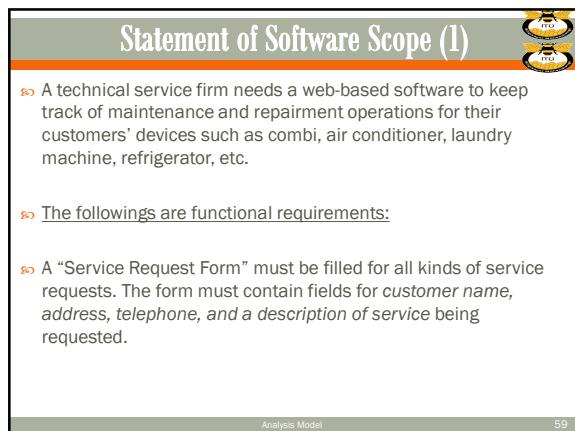
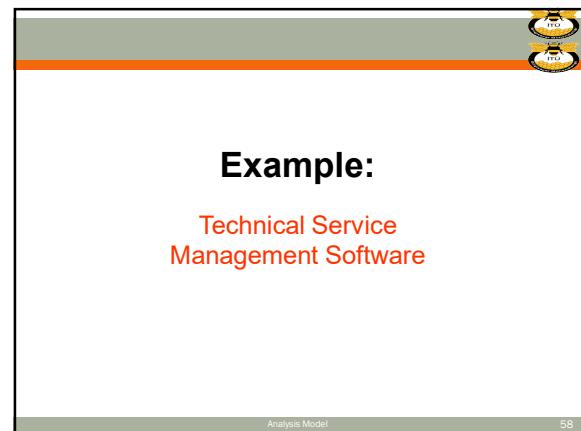
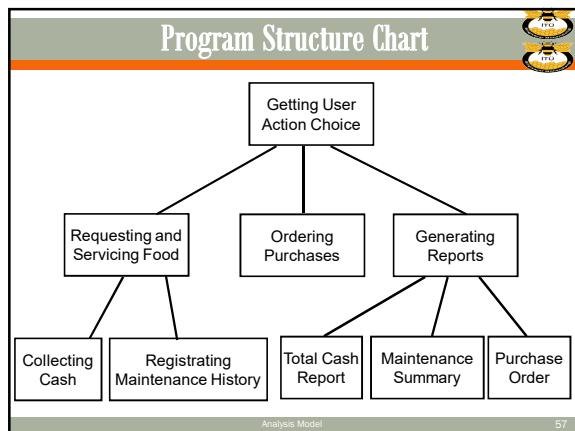
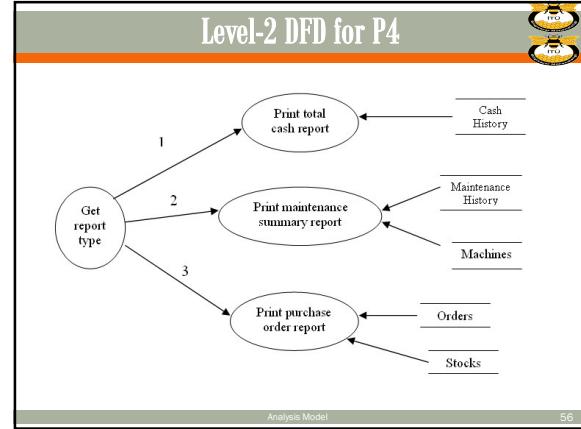
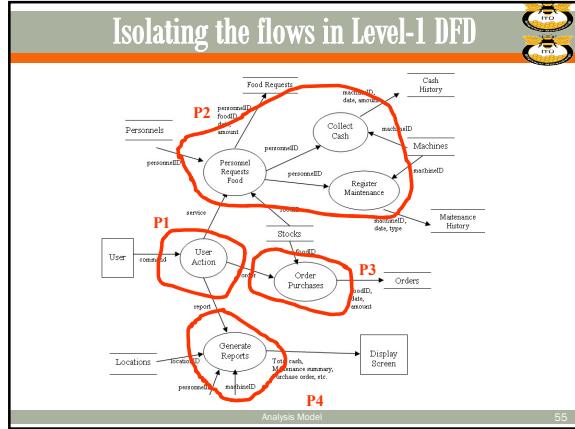


Entities

ENTITY	DATA ITEMS
Locations	Location_ID, Address, Number of consumers, ServicePersonnel_ID
Machines	Machine_ID, Location_ID, Frequency of refilling
Personnels	Personnel_ID, Personnel name
Stocks	Food_ID, Food name, Current amount
Food_Requests	Personnel_ID, Food_ID, Date of request, Amount of request, Returned amount
Cash_Collection_History	Machine_ID, Date of collection, Amount of cash
Maintenance_History	Machine_ID, Date of maintenance, Type of maintenance
Purchase_Orders	Order_ID, Food_ID, Date of order, Amount of order

Analysis Model 52





Statement of Software Scope (3)

- » The manager will assign a service request task to an available technician.
- » For each service request the followings should be recorded: Device information (device type, brand, model, warranty status, start date, expiration date); Jobs done at service, Spare parts used if any, Billing amount (TL).
- » For customers who has warranty agreement, periodic maintanances will be tracked. For this purpose, a list of devices which are sorted by warranty expiration date should be available.

Analysis Model

61

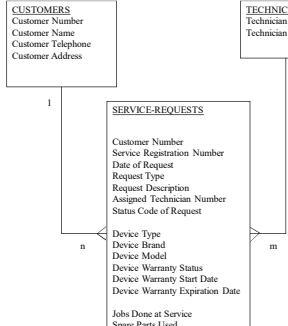
Statement of Software Scope (4)

- » "Service Request Lists" should be available with different criteria:
 - o by service registration number
 - o by customer name
 - o by status code
 - o by device type
 - o by request type
 - o by date of request
 - o by technician name

Analysis Model

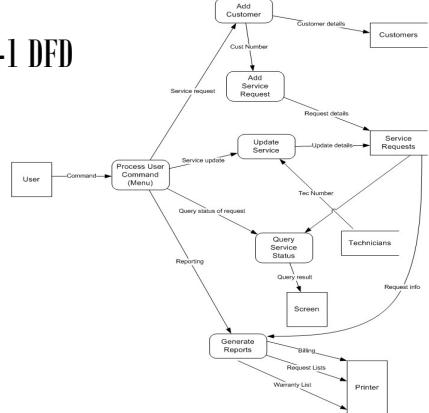
62

Entity Relationship Diagram (ERD)

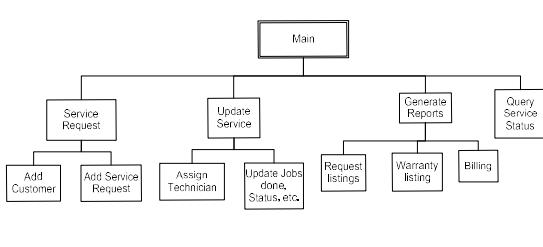


63

Level-1 DFD



Program Structure Chart



Analysis Model

65

1. Requirements Analysis
2. Structured Analysis
 1. Data Model
 2. Functional Model
 3. Behavioural Model
3. Object Oriented Analysis

Object Oriented Analysis

» 6.3 CG

Analysis

Object-Oriented Analysis

- » OOA is a semiformal analysis technique for the object-oriented paradigm
 - There are over 60 equivalent techniques
 - Today, the Unified Process is the only viable alternative
- » During this workflow
 - The classes are extracted
- » Remark
 - The Unified Process assumes knowledge of class extraction

The Analysis Workflow

- » The analysis workflow has two aims
 - Obtain a deeper understanding of the requirements
 - Describe them in a way that will result in a maintainable design and implementation

The Analysis Workflow (contd)

- » There are three types of classes:
 - Entity classes
 - Boundary classes
 - Control classes

The Analysis Workflow (contd)

- » Entity class
 - Models long-lived information
- » Examples:
 - Account Class
 - Investment Class

The Analysis Workflow (contd)

- » Boundary class
 - Models the interaction between the product and the environment
 - A boundary class is generally associated with input or output
- » Examples:
 - Investments Report Class
 - Mortgages Report Class

The Analysis Workflow (contd)

- » Control class
 - Models complex computations and algorithms
- » Example:
 - Estimate Funds for Week Class

UML Notation for These Three Class Types

↳ Stereotypes (extensions of UML)

The diagram shows three UML stereotypes: Entity Class (represented by a circle with a vertical line through it), Boundary Class (represented by a circle with a horizontal line through it), and Control Class (represented by a circle with a curved line through it).

Entity Class **Boundary Class** **Control Class**

Figure 13.1

Extracting the Entity Classes

↳ Perform the following three steps incrementally and iteratively

- Functional modeling
 - Present scenarios of all the use cases (a *scenario* is an instance of a use case)
- Class modeling
 - Determine the entity classes and their attributes
 - Determine the interrelationships and interactions between the entity classes
 - Present this information in the form of a *class diagram*
- Dynamic modeling
 - Determine the operations performed by or to each entity class
 - Present this information in the form of a *statechart*

The Initial Class Diagram: MSG Foundation

↳ The aim of entity modeling step is to extract the entity classes, determine their interrelationships, and find their attributes

↳ Usually, the best way to begin this step is to use the two-stage noun extraction method

Noun Extraction: MSG Foundation

↳ Stage 1: Describe the information system in a single paragraph

- Weekly reports are to be printed showing how much money is available for mortgages. In addition, lists of investments and mortgages must be printed on demand.

Noun Extraction: MSG Foundation (contd)

↳ Stage 2: Identify the nouns in this paragraph

- Weekly reports are to be printed showing how much money is available for mortgages. In addition, lists of investments and mortgages must be printed on demand.

↳ The nouns are report, money, mortgage, list, and investment

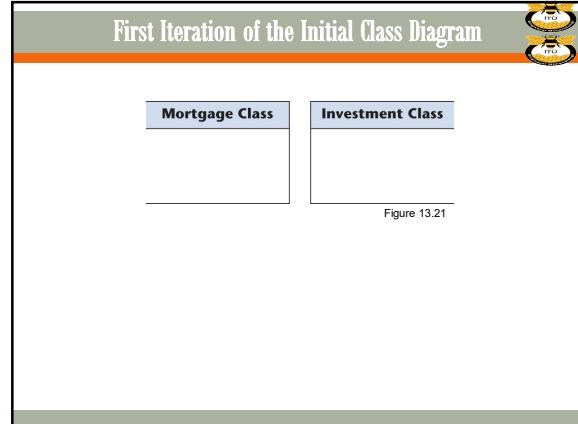
Noun Extraction: MSG Foundation (contd)

↳ Nouns report and list are not long lived, so they are unlikely to be entity classes (report will surely turn out to be a boundary class)

↳ money is an abstract noun

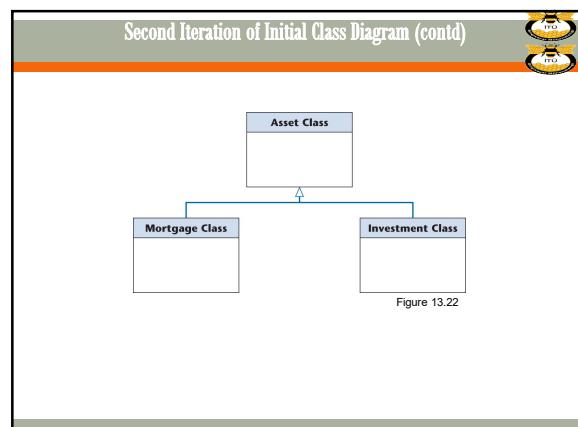
↳ This leaves two candidate entity classes

- **Mortgage Class** and **Investment Class**



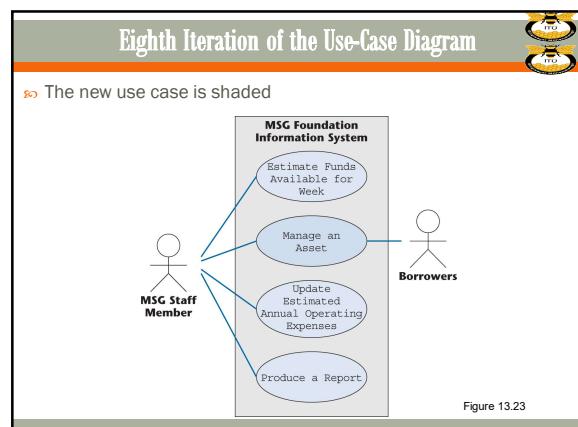
Second Iteration of the Initial Class Diagram

- Operations performed on the two entity classes are likely to be very similar
 - Insertions, deletions, and modifications
 - All members of both entity classes have to be printed on demand
- Mortgage Class and Investment Class should be subclasses of a superclass called Asset Class



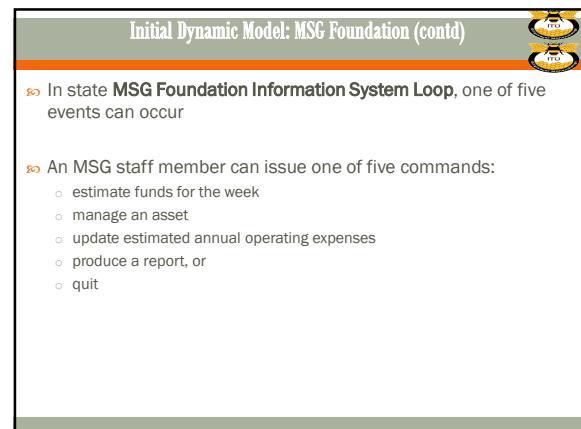
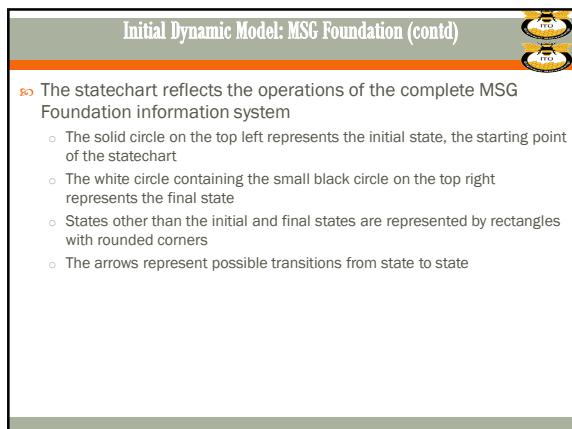
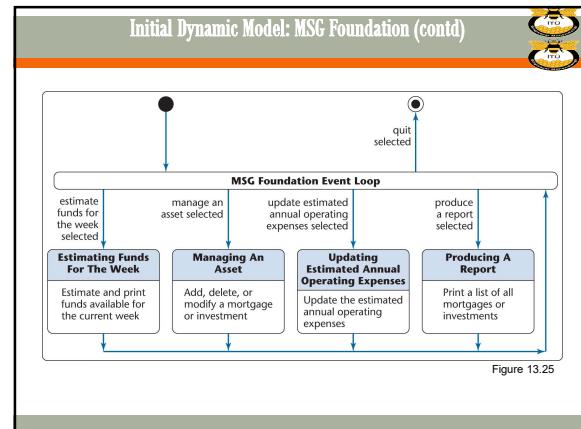
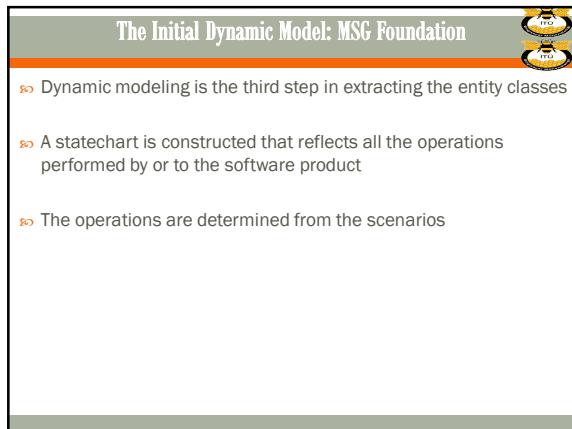
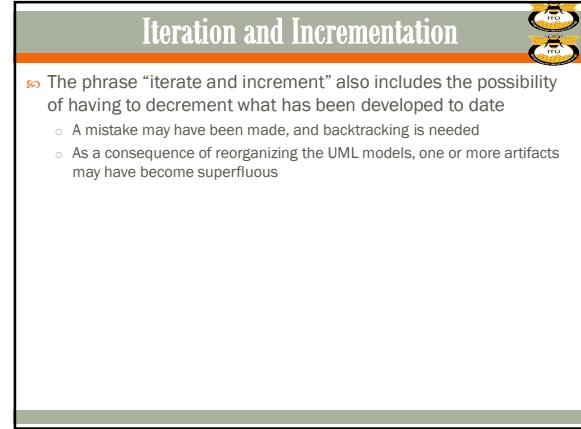
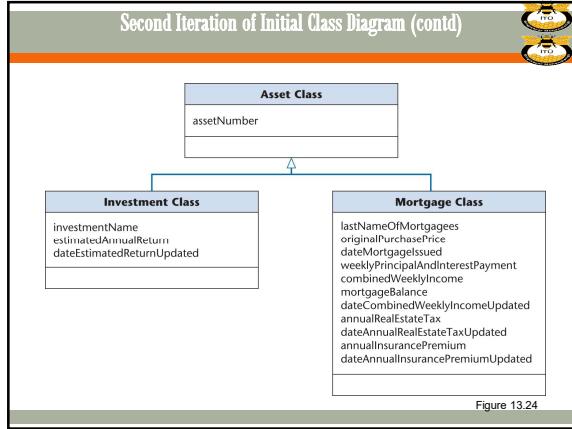
Back to the Requirements Workflow

- The current five use cases include Manage a Mortgage and Manage an Investment
- These two can now be combined into a single use case, Manage an Asset



Initial Class Diagram: MSG Foundation (contd)

- Finally, we add the attributes of each class to the class diagram
 - For the MSG Foundation case study, the result is shown on the next slide
- The empty rectangle at the bottom of each box will later be filled with the operations of that class



Initial Dynamic Model: MSG Foundation (contd)

- » These possibilities are indicated by the five events
 - estimate funds for the week selected
 - manage an asset selected
 - update estimated annual operating expenses selected
 - produce a report selected, and
 - quit selected
- » An event causes a transition between states

Initial Dynamic Model: MSG Foundation (contd)

- » An MSG staff member selects an option by clicking on the menu

Click on your choice:

[Estimate funds for the week](#)

[Manage an asset](#)

[Update estimated annual operating expenses](#)

[Produce a report](#)

[Quit](#)
- » This graphical user interface (GUI) requires special software

Initial Dynamic Model: MSG Foundation (contd)

- » Equivalent textual user interface that can run on any computer

MAIN MENU
 MARTHA STOCKTON GREENGAGE FOUNDATION
 1. Estimate funds available for week
 2. Manage an asset
 3. Update estimated annual operating expenses
 4. Produce a report
 5. Quit
 Type your choice and press <ENTER>;

Figure 13.27

Revising the Entity Classes: MSG Foundation

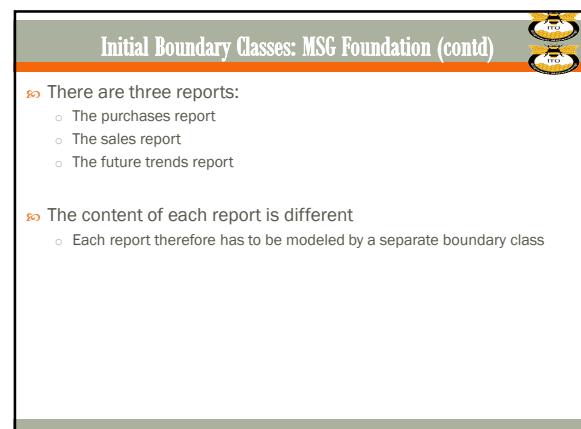
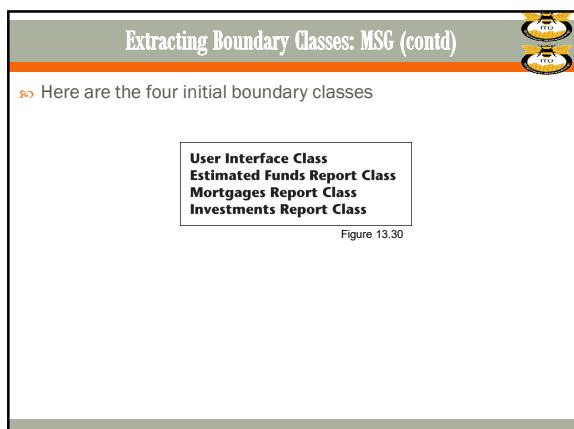
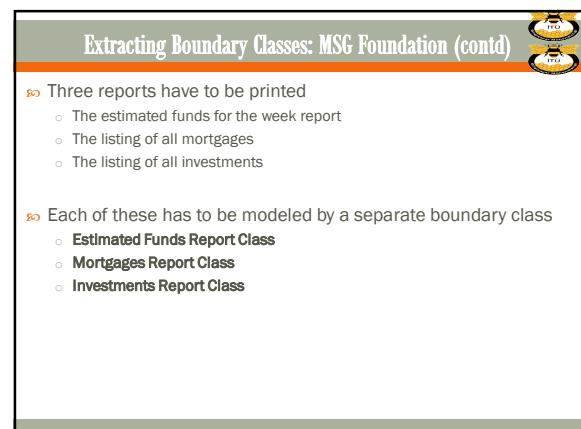
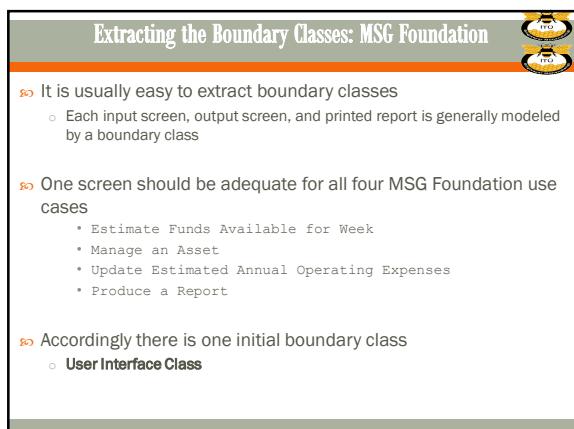
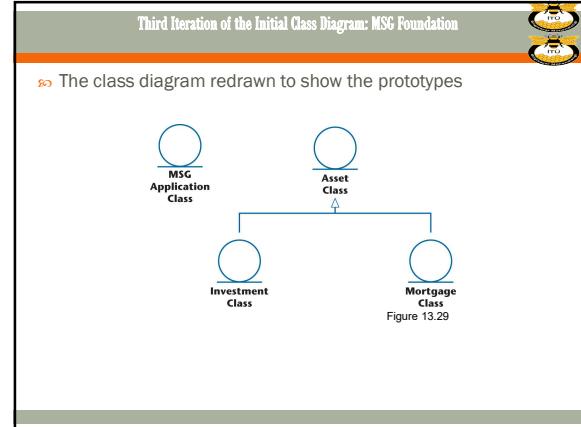
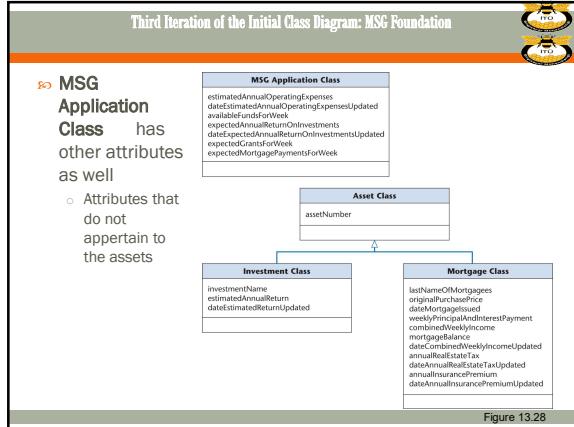
- » The initial functional model, the initial class diagram, and the initial dynamic model are completed
 - Checking them reveals a fault
- » In the initial statechart, consider state **Update Estimated Annual Operating Expenses** with operation **Update the estimated annual operating expenses**
 - This operation has to be performed on the current value of the estimated annual operating expense

Revising the Entity Classes: MSG Foundation (contd)

- » But where is the value of the estimated annual operating expenses to be found?
- » Currently there is only one class (**Asset Class**) and its two subclasses
 - Neither is appropriate for storing the estimated annual operating expenses

Revising the Entity Classes: MSG Foundation (contd)

- » The only way a value can be stored on a long-term basis is as an attribute of an instance of that class or its subclasses
- » Another entity class is needed for storing the estimated annual operating expenses
 - **MSG Application Class**



Extracting the Control Classes: MSG Foundation

- Each computation is usually modeled by a control class
- The MSG Foundation case study has just one
 - Estimate the funds available for the week
- There is one initial control class

Estimate Funds for Week Class

Figure 13.31

Extracting the Control Classes: MSG Foundation

```

control class
Estimate Funds for Week Class
+ <static> compute () : void

public static void compute(EstimatedFunds)
{
    // method computes the estimated funds available for the week
}

float expectedWeekInvestmentReturn;
float expectedTotalWeeklyNetPayments = (float) 0.0;
float totalPrincipalAndInterestPayments;
float grossWeeklyIncome;
float estimatedFunds = (float) 0.0;

Create an instance of an investment record;
Investment inv = new Investment();
Create an instance of a mortgage record;
Mortgage mort = new Mortgage();
mort.setTotalMonthlyPayments(0);

expectedWeekInvestmentReturn = inv.totalWeeklyReturnOnInvestment();
inv.setExpectedTotalWeeklyNetPayments((float) 14.0);
expectedTotalWeeklyNetPayments = mort.totalMonthlyNetPayments();
// now compute the estimated funds for the week
estimatedFunds = inv.getExpectedAnnualOperatingExpenses() / (float) 52.0;
+ expectedTotalWeeklyNetPayments;

// note this value is at the appropriate location
MSGApplication.setEstimatedFundsForWeek (estimatedFunds);
// computeEstimatedFunds

```

Use-Case Realization: The MSG Foundation Case Study

- The process of extending and refining use cases is called use-case realization

Figure 13.32

Use-Case Realization (contd)

- The realization of a specific scenario of a use case is depicted using an interaction diagram
- Either a sequence diagram or collaboration diagram
- Consider use case Estimate Funds Available for Week
- We have previously seen
 - The use case
 - The description of the use case

Figure 13.33

Estimate Funds Available for Week Use Case

- Use-case diagram

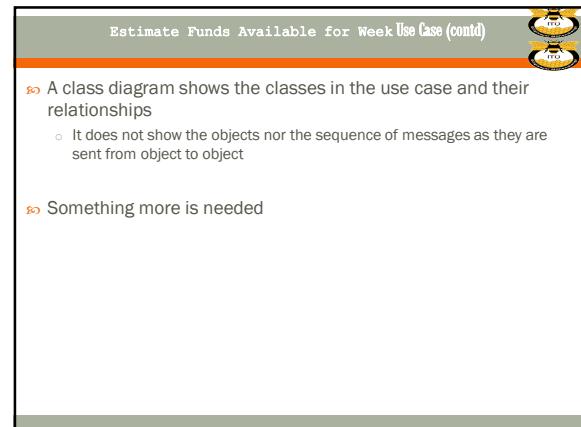
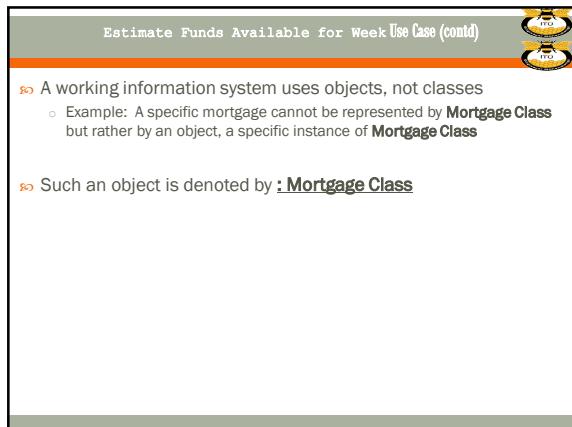
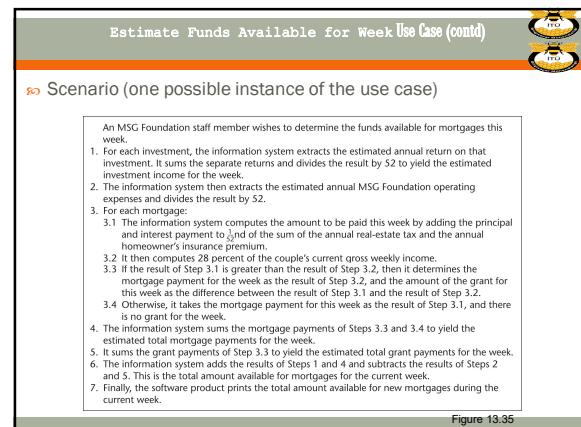
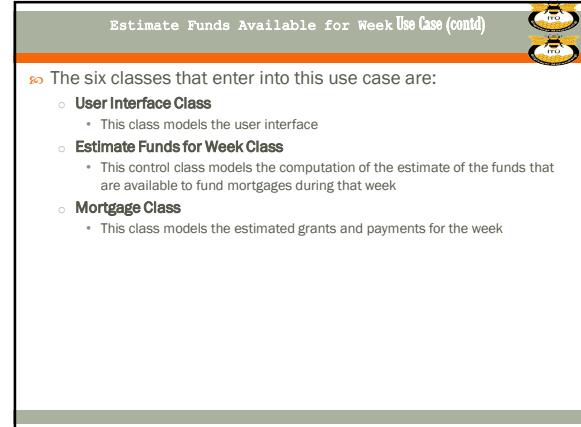
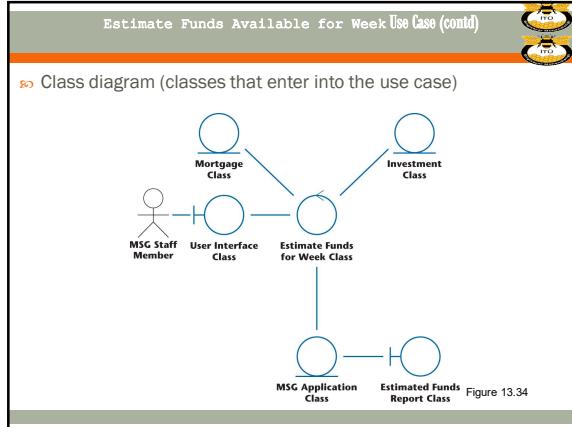
Figure 13.32

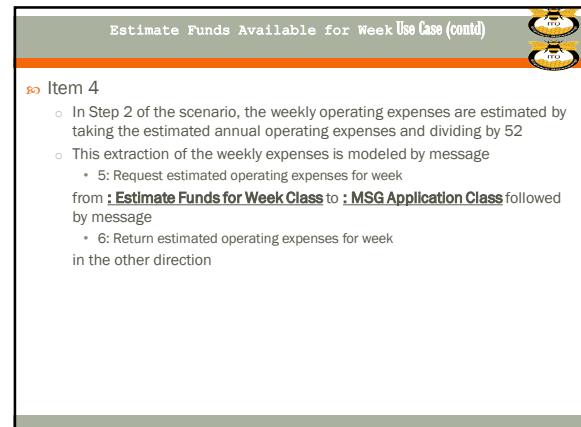
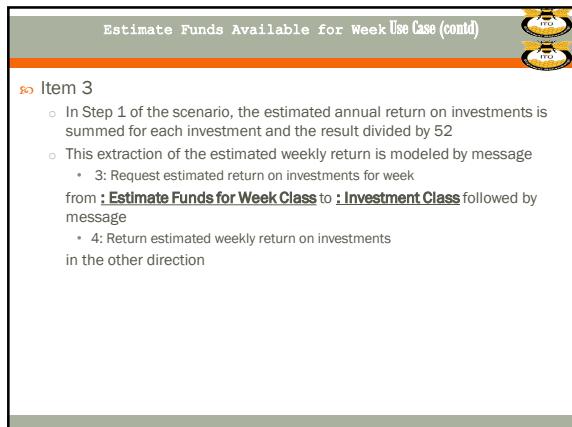
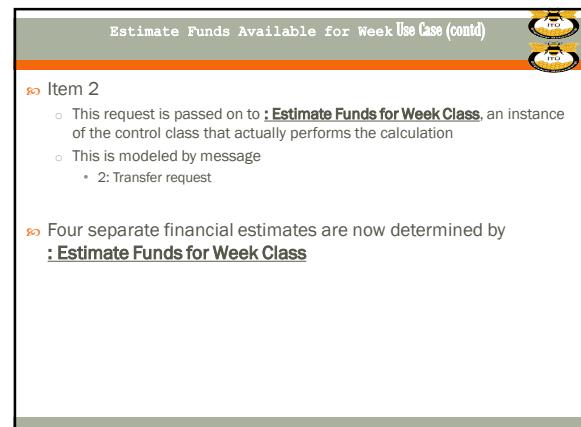
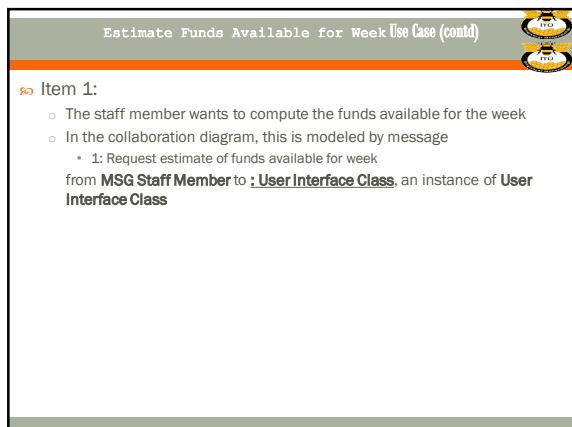
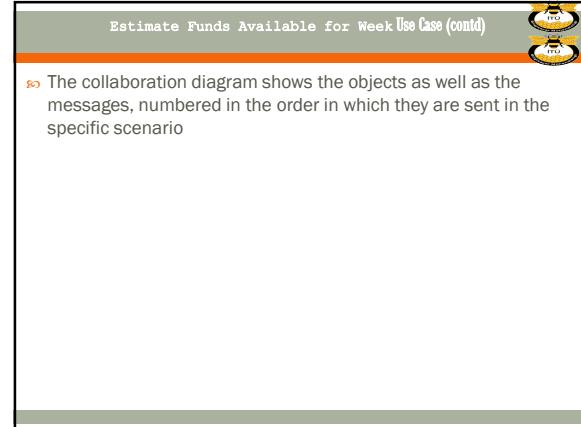
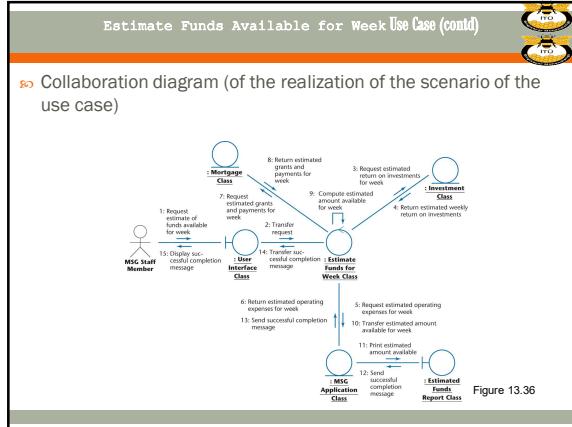
Estimate Funds Available for Week Use Case (contd)

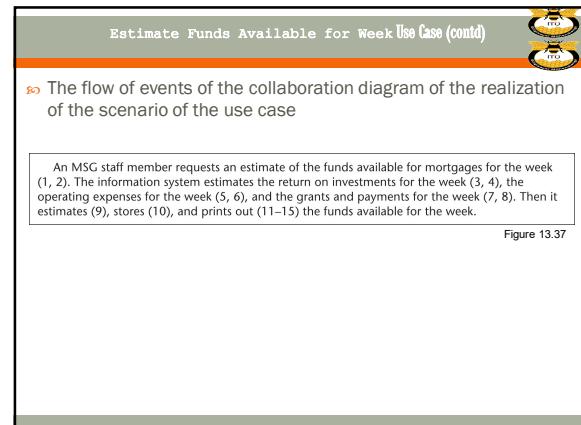
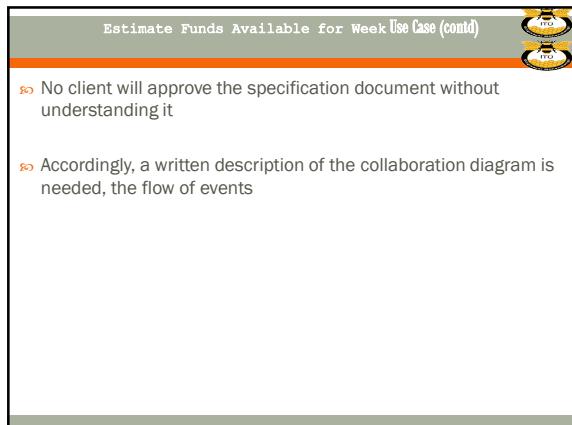
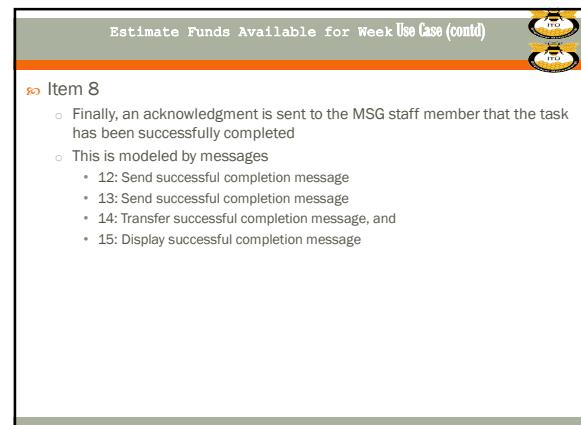
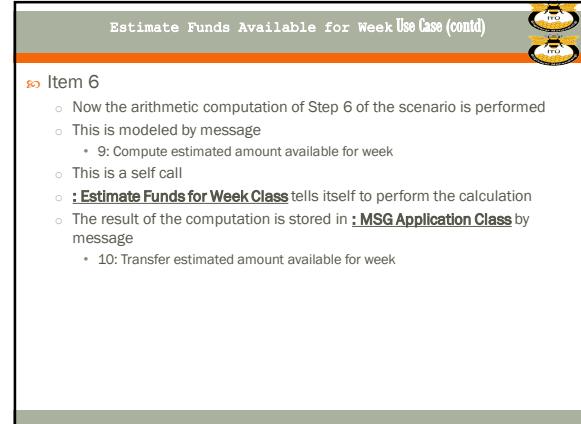
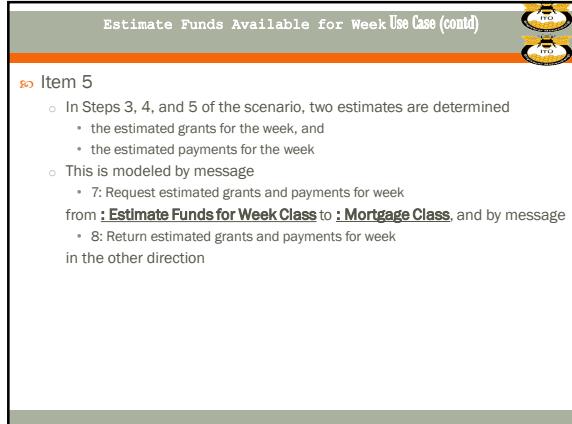
- Description of use case

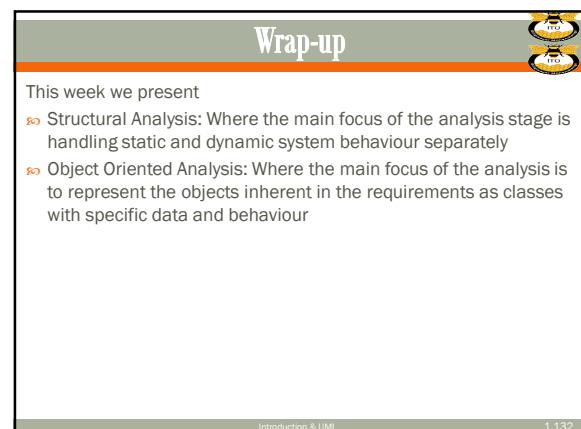
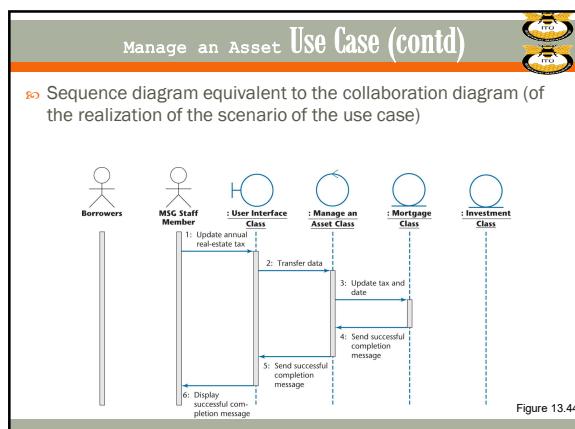
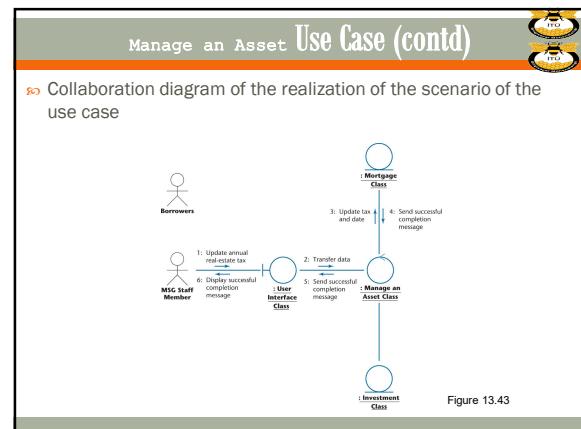
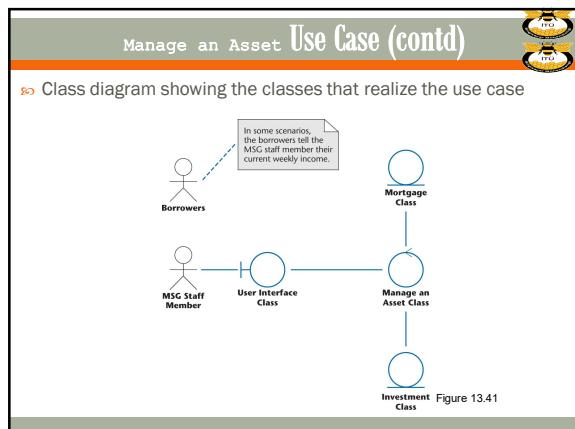
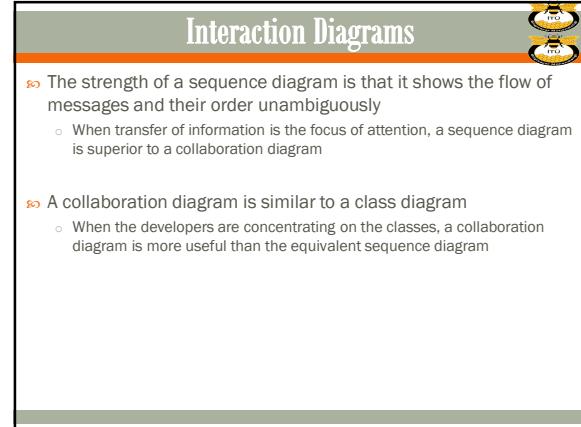
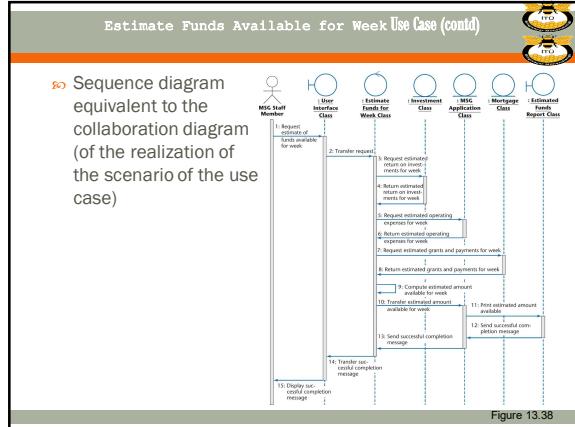
Brief Description
The Estimate Funds Available For Week use case enables an MSG Foundation staff member to estimate how much money the Foundation has available that week to fund mortgages.
- Step-by-Step Description
 - For each investment, extract the estimated annual return on that investment. Summing the separate returns and dividing the result by \$2 yields the estimated investment income for the week.
 - Determine the estimated MSG Foundation operating expenses for the week by extracting the estimated annual MSG Foundation operating expenses and dividing by 52.
 - For each mortgage:
 - The amount to be paid this week is the total of the principal and interest payments and 3% of the sum of the annual real-estate tax and the annual homeowners insurance premium.
 - Compute 28 percent of the couple's current gross weekly income.
 - If the result of Step 3.1 is greater than the result of Step 3.2, then the mortgage payment for this week is the result of Step 3.2, and the amount of the grant for this week is the difference between the result of Step 3.1 and the result of Step 3.2.
 - Otherwise, the mortgage payment for this week is the result of Step 3.1, and there is no grant this week.
 - Summing the mortgage payments of Steps 3.3 and 3.4 yields the estimated total monthly payments for the week.
 - Summing the grant payments of Step 3.3 yields the estimated total grant payments for the week.
 - Add the results of Steps 1 and 4 and subtract the results of Steps 2 and 5. This is the total amount available for mortgages for the current week.
 - Print the total amount available for new mortgages during the current week.

Figure 13.33









Next Week

We will be covering *Architectural Models and Model Driven Engineering!!!*

Introduction & UML 1.133