Lecture 13

Revision

Examination

- IMS1002/CSE1205 Exam
- Check the exam timetable for time and location!
- 3 hour duration
- Worth 60% of subject assessment – marked out of 100 and converted
- You must obtain at least 40% of the mark for this exam (24/60), as well as 40% in each assessable exercise to pass the subject

Examination

- Six questions
  - Five short answer definitions – 20 marks
  - ER modelling – 25 marks
  - Normalisation – 20 marks and 10 marks
  - Medium descriptive essay – 10 marks
  - Long descriptive essay – 15 marks

Examination

- Examination Strategy
  - Marks reflect the amount of effort expected
  - 3 hours (180 minutes) = 100 marks
  - 1.8 minutes per mark eg 3 marks = 5.4 mins
    5 marks = 9 mins
  - Use the allowable time wisely – let it guide the length and content of your answer
  - Don’t spend more time on a question than is available – return to it if you finish another question early

- Examination Strategy
  - Read the questions carefully
  - Make sure you understand what is being asked – don’t answer the wrong question
  - A complete answer has three components
    - A definition
    - An explanation
    - An example
  - Only use bullet points when asked for a list – discussion requires narrative
Review

- Systems Analysis
  - Nature of the SDLC
    - Orderly approach, manageable components
  - Initiation
    - Is the project worth doing?
    - Preliminary investigation, define scope
  - Analysis
    - Define the clients requirements – what?
    - Understand the problem, acceptance criteria
    - Reassess feasibility

Review

- Systems Design
  - How does it happen?
    - Generate alternative design approaches
    - Client selects option
    - Acquire hardware, software
    - Design interfaces, databases, networks
    - Specify integration requirements

Review

- Systems Implementation
  - Build and deliver system
    - Build and modify DBs
    - Build and test programs
    - Prepare users
    - Finalise system and technical documentation
    - Install
    - Alternative development strategies
  - Systems Review
    - What went wrong/right? Why?

Review

- Systems Maintenance
  - Fix it or make it better
    - Corrective - fix errors
    - Adaptive - satisfy changing needs
    - Perfective - enhance performance
    - Preventative - fix potential problems
  - System Cross Life Cycle activities
    - Quality, project management, documentation, ethics

Initiation

- Initiation – preliminary investigation
  - Identify potential projects
  - Classify and rank projects
  - Select project
  - Assess feasibility
    - POSTEL framework
    - Tangible vs intangible costs and benefits, Time value of money (TVM) – NPV, ROI, IRR, Break-even point
    - Plan and present project
    - Baseline Project Plan, Feasibility study

Process Modelling

- Process Modelling - DFD
  - Represents processes that capture, manipulate, store and distribute data
  - Data flows, data stores, processes, sources/sinks
  - Levelling
    - Physical vs logical views of system
Logical vs Physical DFDs

<table>
<thead>
<tr>
<th>Physical DFDs</th>
<th>Logical DFDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>What the system does</td>
</tr>
<tr>
<td>Processes</td>
<td>Essential sequence</td>
</tr>
<tr>
<td>Naming</td>
<td>Underlying data and activities</td>
</tr>
<tr>
<td>Data flows</td>
<td>Only essential inputs and outputs of the processes</td>
</tr>
</tbody>
</table>

Entity Relationship ER Modelling

- **Entity Relationship Modelling - ER**
  - Data “objects” or entities are things about which we wish to store information
  - ER models show the major data objects and the associations between them
  - ER models are useful in the initiation, analysis and design phases

Building a Basic ER Model

- Identify and list the major entities in the system
- Represent the entities by named rectangles
- Identify, draw, name, and quantify relationships
- Indicate mandatory/optional nature of relationships
- Revise for entity subtypes where appropriate

Example ER Model

![Example ER Model Diagram]

Detailed Data Modelling

- Detailed data modelling aims to identify and describe attributes, convert ER models to relations, and to normalise the relations to ensure that they are well structured
- Techniques
  - attribute collection
  - convert ER models to relations
  - Normalisation
  - convert to data structure diagram
  - Data Dictionary

Attributes of Entities

- An attribute is a named property or characteristic of an entity that is of interest to the organisation
- It represents something that the organisation wants to remember about the entity
- Use an initial capital letter followed by lower case letters in naming attributes
- Each attribute name must be unique
Identifying Attributes

- There are three main sources of attributes
  - Data to support essential user functions
    - Item#, ItemDescription, Customer#, QOH
  - Data to support current operations
    - Staff#, UserID, Password, Request#
  - Data to measure performance against objectives
    - ReceiptDate vs DeliveryDate, SaleDate vs PaymentDate

Normalisation

- Normalisation is a process for converting complex data structures into simple, stable data structures in the form of relations

  - Data models consisting of normalized relations
    - are robust and stable
    - have minimum redundancy
    - are flexible
    - are technology-independent (logical)

Steps in Normalisation

- Normalisation to third normal form is accomplished in 3 steps each corresponding to a basic normal form

  - A normal form is a state of a relation that can be determined by applying simple rules concerning dependencies within that relation
  - Each step of the normalisation process is applied to a single relation in sequence so that the relation is converted to third normal form

Normalisation

- First Normal Form 1NF
  - ID primary key, ID repeating groups, remove repeating groups

- Second Normal Form 2NF
  - ID partial dependencies between parts of key, non-key attributes on part of key

- Third Normal Form 3NF
  - ID dependencies between non-key attributes

Steps in Normalisation

- Unnormalised table
- Remove repeating groups
- First Normal Form
- Remove partial dependencies
- Second Normal Form
- Remove transitive dependencies
- Third Normal Form

Normalisation

- During the normalisation process two or more relations with the same primary key may appear

  - The set of 3NF relations must not contain any duplicate data
  - Relations with the same primary key should be merged
Dependencies in Normalisation

- If two attributes ‘A’ and ‘B’ appear in a relation and for each value of attribute ‘A’ there is only ever one value of attribute ‘B’ then attribute ‘B’ is said to be functionally dependent on ‘A’
- For each value of Customer# there is only ever one value of CustomerName

Anomalies in Normalisation

- Create anomaly
  - A record cannot be created until all its attribute values are populated
- Update anomaly
  - An attribute value may have to be updated many times if it appears in many records
- Delete anomaly
  - Deleting a record may remove all reference to an attribute if this is the only record in which it occurs

Human–Computer Interface Design

- The interface is the link between the users and the computer
  - To many users, the interface is the system

Five Commandments of good interface design

- Support “Transportability of Knowledge”
- Be Consistent
- Provide Feedback
- Use Drab Colours
- Make the User Boss

Alternative System Development Strategies

- Traditional SDLC
- Prototyping
- Joint Application Design (JAD)
- Rapid Application Development (RAD)
- Application packages
- Enhancing existing systems
## Stages of Testing

- **Integration test**
- **Acceptance test**
- **Performance test**
- **Function test**
- **Unit (module) test**

**Description:**
- System in use
- Accepted system
- Validated software
- Functioning system
- Integrated modules
- Tested modules

## Integration Testing

- There are a number of strategies that can be used to carry out integration testing:
  - Big-bang testing
  - Incremental Approaches
    - Top-down testing
    - Bottom-up testing
    - Sandwich testing

## Implementation

- Alternative installation methods
  - Direct installation or Abrupt cut-over
  - Parallel installation
  - Phased installation or Staged installation
  - Pilot installation or Single Location conversion

## Maintenance

- Maintenance is NOT just bug-fixing!
  - Corrective maintenance
    - Analysis, design and implementation errors
  - Adaptive maintenance
    - Changing requirements
  - Perfective maintenance
    - Enhance performance, maintainability, usability
  - Preventive maintenance
    - Corrects potential errors and defects

## References