

Ollscoil na hÉireann, Gaillimh
National University of Ireland, Galway

SUMMER EXAMINATIONS 1999

B.E. DEGREE (Electronic Engineering)

APPLIED SOFTWARE ENGINEERING

Prof. D. Bell
Dr. G. Lyons
Mr. D. Chambers

(Time allowed: 3 hours)

Answer **five** questions in all.

Answer at least **two** questions **from each section**.

Use **separate answer books** for each section.

All questions carry **equal marks**.

SECTION A

1. For the Payroll System described in the following narrative:
 - (a). Construct environmental and behavioural models (using DFDs), ensuring that all processes, data flows, stores and external entities are correctly annotated.
 - (b). Convert the behavioural model to a Structure Chart, making use of *coupling* and *cohesion* design criteria to refine the *first-cut* chart. Show all data couples and control flags on the Structure Chart. Indicate any design assumptions used in this conversion and refinement.

Hourly Paid Worker Payroll System

The system is being designed to automate the weekly payroll process within the Personnel department. Each Thursday, the Personnel department receives a Time and Attendance Report computer file from the Manufacturing department. This file contains details of hours worked for all hourly paid employees. The Personnel office validates this report and discards any invalid entries. Based on the validated time report, gross pay is then calculated, using the appropriate pay rates stored in the Employee Payroll file. This file also contains details of the correct tax and insurance deductions for each employee. Employee records are identified by Employee_ID. For each of these, the correct net pay is then calculated, taking account of all calculated deductions. Pay cheques are then created, taking account of the employee net pay and correct employee details, where employee name and address are drawn from the Employee Master file. Finally, pay cheques are formatted and sent to the Pay Cheque Printer for batch printing.

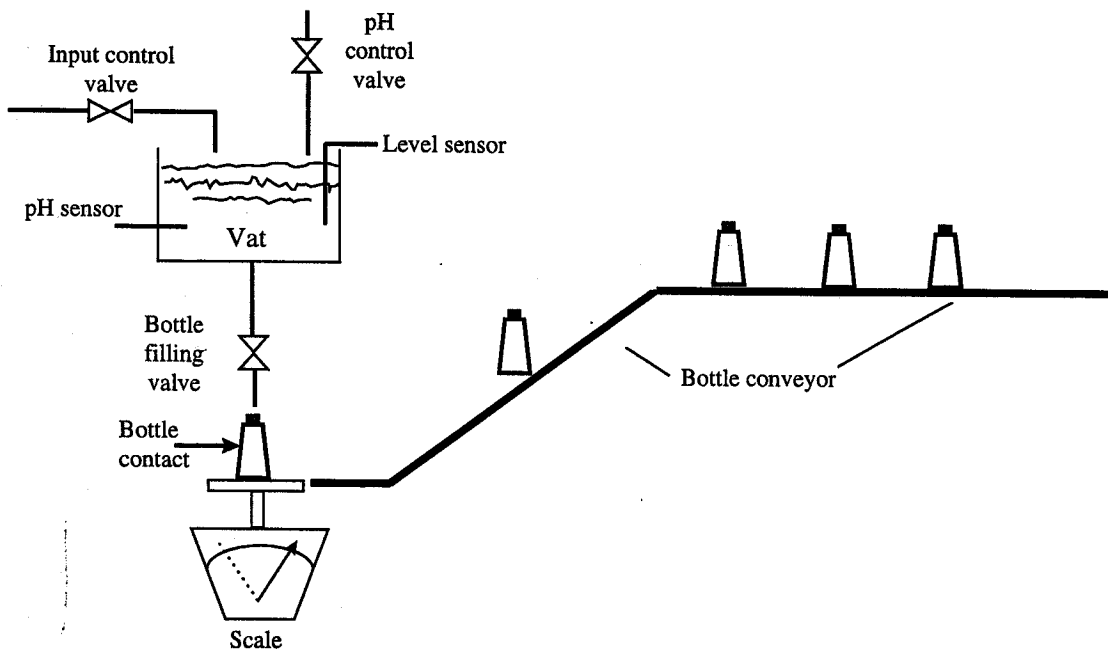
2. (a). Procedural design must specify functional detail unambiguously. What functional design methods are used to specify algorithmic detail? Describe the basic constructs found in structured specification and illustrate your answer using conventional flowcharting notation. Give a clear example for each construct.
- (b). Construct a Decision Table for the manufacturing information requirements articulated in the following statements:
 - The Production Planning Manager must receive an exception report of any deviations of more than 10% from plan, across all products.
 - The Sales Manager must receive a report about any falling behind of plan by more than 5% for product P.
 - The Cost Accountant must receive a copy of all production performance reports for product P.

3. (a) In *Data Analysis and Design*, what common conditions contribute to complexity and redundancy in data structures ? Explain the 3-stage process of *normalisation* leading to 3rd. Normal Form, and illustrate how each stage is achieved in practice by manipulating data entities, attributes and primary keys.
- (b) Convert the un-normalised table shown below into a set of Third Normal Form relations, explaining step-by-step the procedure you used.

Product Code	Product Type	Product Name	Customer Code	Customer Name	Zone	Country	Date last order	Quantity last order
782	Simple	Valve-90	4161	Basaglia	Eur	Switz	12/12/97	1000
			2396	Yang	Asia	Thai	05/01/98	1200
944	Complex	Pump-5C	2279	Wagner	NA	USA	09/12/97	250
			3139	Jones	Eur	UK	10/12/97	100

4. (a). Briefly describe the main limitations of conventional data flow modelling techniques in the analysis and design of real-time systems. Illustrate how the Ward and Mellor extensions to the basic DFD model overcome some of these difficulties.
- (b). For the bottle filling system described in the following schematic and Event List, develop an annotated State-transition Diagram, indicating all appropriate conditions and actions.

Bottle Filling System – Schematic



Bottle Filling System – Event List

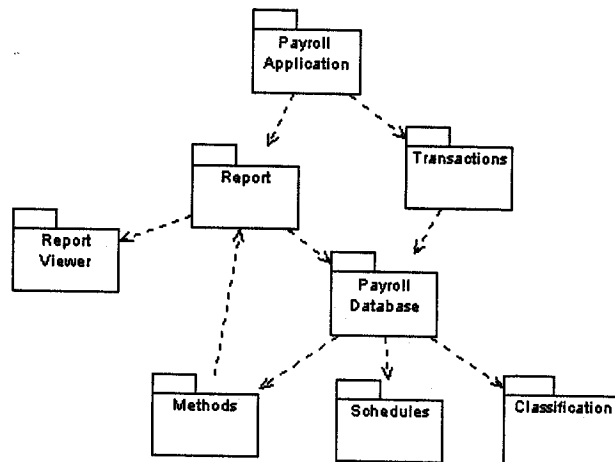
Supervisor enables bottling line
Supervisor disables bottling line
Supervisor sets pH level
Operator turns line ON
Operator turns line OFF
Operator sets new bottle size
Operator removes bottle
Bottle drops into place (bottle contact)
Bottle becomes full (bottle weight)
pH moves out of limits
pH moves within limits
Operator turns conveyor ON
Operator turns conveyor OFF

SECTION B

5. (a) Describe what you would consider to be good attributes in terms of the design of a large software system. Consider in your answer the related concepts of *coupling*, *cohesion*, *interface* and *encapsulation*.
- (b) Describe the concepts of *polymorphism* and *inheritance*. Are there any potential problems with using inheritance?
- (c) Give the full C++ or JAVA class definition for an object that represents a typical (text only) E-Mail message. Show then how you would reuse this class in the definition of a new (derived) class for messages that can include attachments e.g. an attachment could be a MS Word file or a multimedia clip in a supported MIME format.
6. (a) What earlier methods / notations are credited as the main influences on UML.
- (b) Draw a UML use case diagram for the following problem description.
"The system is an ECG monitor. The ECG has a chart printer and a remote display for use by a surgeon. The ECG captures and displays waveforms. Alarms can go off if a problem is detected. The surgeon can configure the display of waveforms on the remote display. The software is upgraded and the sensors are calibrated by a technician."
- (c) Draw a UML interaction (sequence or collaboration) diagram for the following scenario:
"A user updates the value of a threshold on an object of type ControlPanel (instance named centralOffice). If an error count number reported by a transducer is larger than the threshold on the ControlPanel an alarm message is sent to a number of management systems."
7. For the Employee Directory Service described in the following narrative:
"The service is used to hold details on employees within a company and should provide a high level of availability i.e. it should use some form of replication in case an individual copy of the directory service becomes unavailable. Employees within a company should be allowed to register themselves with the service - typically they will use the service to register their Name, Department, E-Mail address, Phone Number and any other relevant details."

As this is an internet based service, it should also include some security features. Specifically it should not allow any access to requests from outside the companies internet domain(s) and should also use the concept of capabilities (security tokens) to further restrict access to the directory itself i.e. when a user registers with the service, the user is returned a capability which allows that user full access to his / her directory entry (including the right to modify or delete the entry). All other users should be returned reduced capabilities which allow only read / search access to directory entries. The only exception is that the designated manager of the service should also have the capability to delete any entries within the service."

- (a) Derive a top level Class Diagram for this system, showing important attributes, methods and links between classes.
 - (b) Should any of the resulting classes be defined as *abstract classes* or *interfaces*?
 - (c) Discuss how the *Proxy* and *Observer* Design Patterns might be used in the design or extension of this system.
8. (a) What is meant by the term "*Software Reflection*" and give examples of where it's used.
- (b) Describe the dependency inversion principle. Illustrate using an example (which includes a class diagram). What are the consequences of applying the principle on small systems?
- (c) What problem does this diagram indicate? How would you solve the problem?



Why would the PayrollDatabase package/component export an abstract interface? What principle does this illustrate?