UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1999

BEng Honours Degree in Computing Part II
MEng Honours Degrees in Computing Part II
BSc Honours Degree in Mathematics and Computer Science Part II
MSci Honours Degree in Mathematics and Computer Science Part II
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute

PAPER 2.1 / MC 2.1

SOFTWARE ENGINEERING – DESIGN I
Friday, May 7th 1999, 4.00 – 5.30

Answer THREE questions

For admin. only:
paper contains 4 questions

Section A *(Use a separate answer book for this Section)*

1a What is a *model* (sometimes called a *binding)* of a schema? (Assume for simplicity that the schema has no base sorts (generic sets), i.e. no symbols in square brackets after the schema name.)

For each of the following schemas, describe what would constitute a model and state, with reasons, how many models it has.

i)

______ Small ______

n: **nat** (**nat** here is the type of natural numbers)

$0 \le n \le 4$

---

ii)

______ Fn ______

f: **nat**→**nat**

$\forall x$: **nat**. $f(x) = x+1$

---

iii)

______ Op ______

ΔSmall

$n' = n+1$

---

iv)

______ Prop ______

$P \subseteq$ **nat**

$\forall x$: **nat**. $(P(x) \rightarrow 0 \le x \le 4)$

---

v)

______ P1 ______

Fn

$P \subseteq$ **nat**

$P(0)$

$\forall x$: **nat**. $(P(x) \rightarrow P(f(x)))$

---

b Let schemas be defined as follows:

______ Var ____________________

n: **nat**	**nat** here denotes the type of natural numbers

______ DecNormal ____________________

ΔVar
d?: **nat**
report!: {normal, error}

---

n = n'+d?
report! = normal

______ DecPre ____________________

DecNormal

Pre: d? $\leq$ n

______ DecError ____________________

ΔVar
d?: **nat**
report!: {normal, error}

---

n = n' < d?
report! = error

Dec = DecNormal $\vee$ DecError

i) Write out how Dec would appear if written out in full without inclusions, Δ or schema disjunction.

ii) Show that DecNormal and DecPre have the same models.

iii) Of the three operation schemas DecNormal, DecPre and Dec, which are *total* operations? Give reasons.

Turn over ...

2a i) What is the distinction between *state schemas* and *operation schemas?*

ii) Explain how the three symbols ', Δ and Ξ are used in operation schemas.

b *In this part of the question, where you are asked to write schemas, you are not required to follow the Z notation strictly. However, your logic should be clear.*

A drinks dispenser serves tins of ginger beer in return for money. You put coins in a slot, and when you have paid enough you can press the "dispense" button to obtain your drink and any change due. At any time before your drink is dispensed you can press a "cancel" button for a full refund of the money you have paid.

A schema *Machine* is given describing the basic parameters of the machine:

```
_____ Machine ________________________________
| capacity: nat          (number of tins when full)
| price: nat             (price per tin)
|_____________________________________________
|
|_____________________________________________
```

(**nat** here denotes the type of natural numbers)

i) Write a state schema *State* for the machine, including *Machine* and also using variables *stock* (number of tins left), *paid* (amount of money inserted so far in the current transaction) and *takings* (amount of money taken previously for drinks dispensed). Include relevant constraints on the variables.

ii) Write operation schemas for the following four responses to button presses, using outputs *tinsout!* (number of tins delivered, 0 or 1) and *change!* (money returned). Use Ξ where appropriate.

DispenseNormal dispenses a tin and change.

DispenseEmpty dispenses nothing because there are no tins left. It does not refund the money paid (*Cancel* does that).

DispenseUnpaid dispenses nothing because not enough money has been paid yet.

Cancel gives a refund.

iii) For your schemas in ii), show that the value *takings* + *stock*price* is left invariant by each operation.

The two parts carry, respectively, 25%, 75% of the marks.

Section B *(Use a separate answer book for this Section)*

3

a Explain the use of *associations* in object modelling.

b Explain the use of *inheritance* in object modelling.

c Draw a class diagram for the following requirements:

> The system maintains information on telephone calls, which may be of three types: local, long-distance, and international. Each call has a caller and callee, which are (distinct) people. A call has a network on which it is established. There are attributes **duration** and **charge_rate** of each call. There are operations **start** and **finish** to begin and end a call.

The three parts of this question carry, respectively, 15%, 15% and 70% of the marks

[Turn over ...

4

a List and explain three of the principles of software engineering.

b Draw a state transition diagram for the controller of the following system:

> The system is concerned with the operation of an entry manway in the side of a large tank in a milk processing plant (Figure 1). The system raises an alarm if the manway is open while the tank is in operation: there is a sensor for the manway and one for operation of the tank. There are two operations **raise_alarm** and **cancel_alarm** on the alarm. The state machines of these three components are shown in Figure 2. The context diagram of the system is shown in Figure 3.

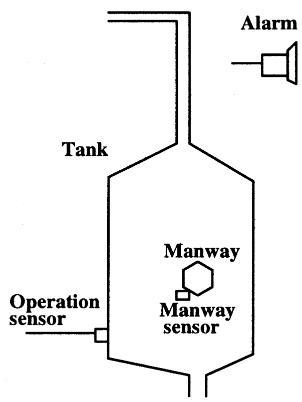

Figure 1: Tank system components

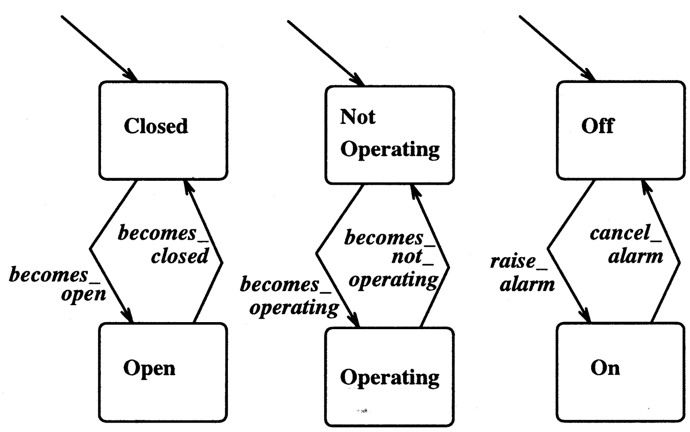

Figure 2: State machines of tank sensors and actuator

The controller statechart should raise an alarm if the manway is open and the tank operating, and cancel the alarm if either condition does not hold.

The two parts of this question each carry 50% of the marks.

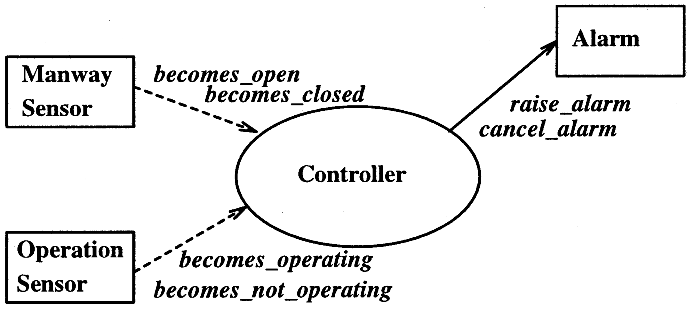

Figure 3: Tank system context diagram

[End of paper