UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2003

MSc in Computing for Industry
MEng Honours Degree in Information Systems Engineering Part IV
MSci Honours Degree in Mathematics and Computer Science Part IV
MEng Honours Degrees in Computing Part IV
MSc in Advanced Computing
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
This paper is also taken for the relevant examinations for the Associateship of the Royal College of Science

PAPER C475=I4.16

SOFTWARE ENGINEERING - ENVIRONMENTS

Thursday 1 May 2003, 10:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators not required

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

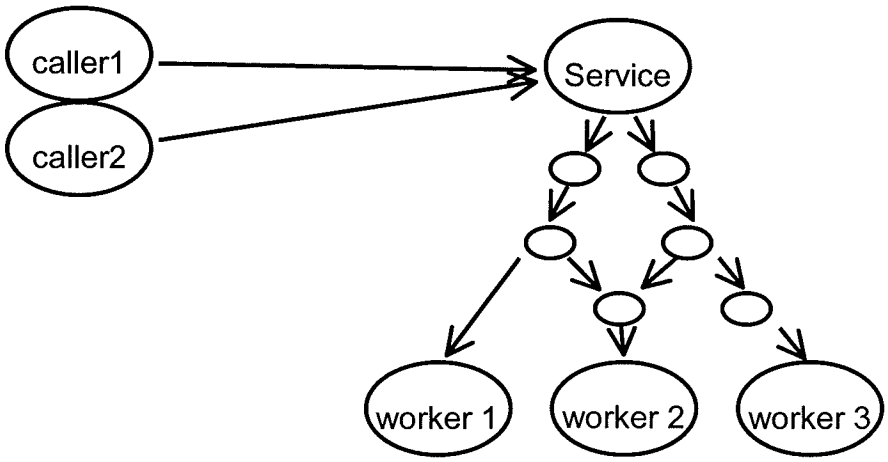
1a **(Aspect Oriented Programming)** Ever since the 1970s there have been program design techniques that dealt with separation of concerns. A recent one is Aspect Oriented Programming.

i) Describe briefly (one sentence each) three earlier techniques.

ii) What does AOP promise to provide that is missing from these earlier techniques?

iii) Give three examples of problems amenable to AOP. For each of your examples what would you have as an aspect and why?

b



Consider a problem where there are workers who provide sophisticated services for callers. The figure above shows this situation. Dependent on the caller the services have to be tailored with respect to the caller's capabilities, how much the caller is to be charged and what form the result should be provided in.

i) Describe two concrete examples that follow this pattern.

ii) Write (in pseudo-code or Aspect J) an abstract aspect `CapabilityChecking` that provides pointcuts for service `invocation`, `workPoints`, and `perCallerWork`. When should `perCallerWork` be called (eg `before`, `after` etc)? You can assume that `Service` provides a method `doService(String)`, `Worker` provides a method `doTask(Task)` and `Worker` also provides a method `checkCapabilities(Caller)`.

The two parts each carry 50% of the marks.

2 **(Program Maintenance)**You have started a job at "Backups 'R' Business" a software house that develops archival software for a variety of operating systems. Your job is to answer the customer queries phone line and solve problems if they are small, or pass them on to the maintenance team if they are more substantial. After working for a month you are completely stressed out. Not only does the phone ring continuously with unhappy customers who have not succeeded in retrieving their files, but the archival software seems completely disorganised. There are many versions of the different source and object code and the only way that you can figure out what a customer is running is to try a variety of programs and see which one behaves the way the customer's product behaves. You cannot always find matching source and object code. You sometimes fix a bug in one version of the product only to get a call later on from another customer who is running a different version on another platform with the identical complaint. There is almost no documentation and the code is not commented. Programmers frequently change code without notifying anyone because they don't have the time.

a One of the common types of problems that you are told about is normally termed "DLL Hell".

i) What are DLLs? What are the benefits of building a system using DLLs?

ii) Describe two different scenarios that lead to "DLL Hell".

b Some of the archiving software is over 15 years old.

i) What are the problems specifically associated with such old code?

ii) How would you go about solving these problems?

c Programs that are monolithic comment-less spaghetti may work satisfactorily for years but may not be maintainable. You suggest to your employers that they convert all their codebase, written in C++ to Java.

i) The C++ code contains global variables and code (C++ is an extension to C). Classes are in more than one file. It uses pointers and pointer arithmetic. Some of the recursive methods have reference parameters. The file handling is different. The C++ programs use multiple inheritance. For each of these features say how you would translate into Java.

ii) After the translation was done the programs run much too slowly. Give two reasons why this might happen and for each discuss what can be done to improve matters.

The three parts carry 25%, 35% and 40% of the marks, respectively.

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

3 **(Object modeling in the presence of constraints)** You are asked to model, in Alloy, the core state components and key operations of the transaction processes that would occur in a restaurant. **Note**: only declare signatures that provide structure for your modeling purposes.

a Write appropriate signatures and constraints (where applicable) for

i) Staff, which is either Waiter, KitchenStaff, BarStaff or AdminStaff; a member of staff has a name, salary, login, password, and shifts. A waiter also has orders to deal with.

ii) Bill, which has a date, time, and total; a bill is associated to an order, a set of payments, and a set of billItems; a bill may be verified.

iii) OrderItem, has a name and a price, is associated with an order, may be vegetarian, and may have variations.

iv) Order, which may have a bill, and has a date, time, name, waiter, and consists of OrderItems. An order also has an OrderStatus, which is either Archived, Complete, Incomplete or PaidFor.

v) State, which has waitersOnDuty.

vi) Table, which has a number and a set of orders.

b If any of the following are already constrained by your signature, please indicate where; otherwise, write facts in Alloy to express

i) WaitersIdentity : the names of waiters uniquely identify their remaining attributes.

ii) NoConflictingOrders : waiters that differ in some of their attributes cannot share an order.

c Write specifications of transactions and a simulation for your Alloy model and specify in which scopes you would run them. For transactions, make sure that you state appropriate preconditions and postconditions:

i) Simulation, whose solutions --- if they exist --- display a snapshot with two administrative staff, two kitchen staff, one bar staff, three waiters, six tables, five orders such that five out of **all** possible order items have at least one variation.

ii) CancelAnOrder : has two states s and s', a nameOfWaiter, and an order o as parameters; its intent is that order o gets cancelled by nameOfWaiter in state s resulting in state s'.

iii) WaiterCreatesOrder : has a table t, an order o, a waiter w, and a name n as parameters; its intent is to create the order o with name n for table t by waiter w.

d Write assertions in Alloy that you would use to analyze your model. In each case, indicate the scopes in which you would check these assertions. Prove that the assertion in question is true/false in your model:

i) OrdersHaveATable;

ii) CreatedOrdersStatus: orders that have just been created are neither PaidFor, Complete nor Archived;

iii) WaitersDontHaveArchivedOrders.

The four parts carry, respectively, 25% of the marks.

4 **(The Alloy constraint language) – answer part a and *either* part b or part c.**

a **(Language features & capabilities) answer any 7 of the 10 parts.**

For **seven** of the concepts stated below (1) define it and give an example of it in computer science in general; (2) state whether, and to what degree, the concept is expressible in Alloy; (3) if applicable, show an example of the concept's application in Alloy; and (4) discuss advantages and disadvantages of having such a concept in a modeling language:

i) subtypes

ii) inheritance of structure

iii) multiplicity constraints

iv) modularity

v) higher-order constraints

vi) recursive datatypes

vii) polymorphism

viii) invariant checking

ix) name types (i.e. two types are the same if they have the same name)

x) structural types (i.e. two types are the same if they have the same structure).

b **(Invariants and safety properties) – answer either this part or part c**

A constraint c is an invariant of a transaction t iff whenever t is executed in any state satisfying c, its successor states satisfy c as well.

A model is safe with respect to a constraint c iff no intial state of the model can reach a state that satisfies c. (Constraint c is then a safety property of the model.)

Discuss in detail how these concepts (invariant & safety property) relate to each other and whether, and under what circumstances, they differ.

c **(Small scope-hypothesis) – answer either this part or part b**

i) Define the small-scope hypothesis, as formulated for the modeling language Alloy and its constraint analyzer.

ii) Explain the practical significance and impact of the small-scope hypothesis on modeling and analysis activities.

iii) Give an informal example of a model in which the small-scope hypothesis fails.

iv) Discuss whether one could conceivably write a modeling and analysis tool whose language is at least as expressive as that of Alloy such that the small-scope hypothesis is no longer needed.

*The **two** parts carry, respectively, 67% and 33% of the marks.*