UNIVERSITY OF LONDON

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1996

MEng Honours Degrees in Computing Part IV
MSc Degree in Foundations of Advanced Information Technology
for Internal Students of the Imperial College of Science, Technology and Medicine

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

PAPER 4.74

DISTRIBUTED ARTIFICIAL INTELLIGENCE

Monday, April 29th 1996, 10.00 - 12.00

Answer THREE questions

For admin. only: paper contains
4 questions
5 pages (excluding cover page)

1a Briefly describe the key features of the *contract net* protocol for allocating tasks to a network of agents/problem solvers. Include descriptions of:

i) the role of a manager

ii) the role of a potential contractor

iii) the content of a contract announcement

iv) when a potential contractor is allowed to submit a bid

v) the use of *node available* messages.

b What are the different circumstances that can result in a manager not receiving any bids before the bid submission cutoff time? In one of these circumstances, there is no point in the manager re-sending the contract announcement to the same set of potential contractors. Why? Suggest possible changes or extensions to the contract net protocol to avoid this problem.

c In the contract net it is assumed that an agent concurrently ranks received contract announcements whilst it is processing a task from a previously awarded contract. However, it is assumed to be working on only *one* task at a time.

Sketch an implementation of a contract net agent as a collection of processes in the April programming language that allows an agent to process several tasks concurrently. In addition, *at any stage* during its processing of any one of these tasks, the agent should be able, concurrently, to try to find subcontractors for one or more subtasks of the task. You do not need to give any April program code. In such a regime, when do you think an agent should be allowed to bid for a new contract?

The three parts carry, respectively, 40%, 30%, 30% of the marks.

2a What is meant by *coherence* in a distributed problem solver?

b In a simple distributed blackboard problem solving architecture, cooperation might be achieved by having each problem solver send certain types of hypotheses to one or more other problem solvers. The hypothesis receiving problem solvers then add them to their local blackboards, and treat the hypotheses as though they had been locally generated.

i) What problems might arise using this simple model?

ii) What benefit might result if a problem solver could explicitly request a hypothesis of a certain form?

iii) What problems or benefits might arise from allowing a problem solver to request raw data as well as hypotheses?

c In a series of experiments with the Distributed Vehicle Monitoring Testbed (DVMT), extra features were progressively added to the simple distributed blackboard problem solving architecture described in part b to increase coherence.

i) Briefly describe these extensions, saying what benefits and drawbacks each may have had.

ii) Suggest other ways, or further extensions, that might lead to improved coherence.

The three parts carry, respectively, 20%, 20%, 60% of the marks.

Turn over ...

3a Briefly explain the role of a *matchmaker* in a distributed information system.

b Explain the role of the KQML performatives: *ask-about, advertise, recommend, subscribe.*

c The program below is an April program which when forked as a process can act as a simple matchmaker.

The program assumes that a KQML message of the form:

```
(performative-name :f1 v1 :f2 v2 .... :fk vk)
```

is represented as the April record:

```
[performative_name, [[f1,v1],[f2,v2],...,[fk,vk]]]
```

The program:

```
keypairs::=[symbol,any][];
MM(){
[handle?advertiser,symbol?qtype,symbol?ontology,symbol[]?topics][]?Ads:= [];
repeat {

[advertise,keypairs?KVPairs]::
      [content,[ask_about,keypairs?askVPs]] in KVPairs ->
            Ads := Ads <> [[sender, ask_about,
                                    lookup(ontology,askVPs),
                                    lookup(content,askVPs)]]


|
[recommend,keypair[]?FVPairs ::
                  [content,[ask_about,keypairs?askVPs]] in FVPairs ->

      if [handle?A,ask_about,symbol?Ont,symbol[]?Topics]::
                        lookup(ontology,askKPs) ==Ont and
                        lookup(content,askKPs) in Topics
            in Ads
      then
            [reply,[[in_reply_to_of,lookup(reply_with,FVPairs),
                  [ontology,Ont]
                  [content,A]] >> replyto
      else  [reply,[[in_reply_to, lookup(reply_with,FVPairs)],
                  [content,unknown]] ] >> replyto
   } until quit
   };

lookup(symbol?S,keypairs?L) -> any?valof{
      if [S,V] in L then valis V else valis no_value};
```

i) Give the April message that an agent would send to this matchmaker if it wanted to advertise the fact that it could be asked about any topic on the topic list [pentium,windows95] using the ontology: computing1.

ii) Give the April message that an agent would send to this matchmaker if it wanted to find the identity of an agent that it could ask about the topic: windows95 using the ontology: computing1.

iii) Modify the program so that it can also handle subscribe messages where the content is an ask-about containing a single topic and an associated ontology. Correct syntax is not important. Make up your own message format for the responses to a subscribe.

The three parts carry, respectively, 20%, 30%, 50% of the marks.

Turn over ...

4a AGENT0 implements a simple model of an agent with an internal state comprising a set of *beliefs*, *capabilities* and *commitments*.

i) Briefly describe the three state components and the role that they play.

ii) How are beliefs and commitments updated?

iii) What can cause a commitment to be abandoned.

b AGENT0 agents only accept three forms of messages.

i) What are these message forms?

ii) What extra functionality, and what, if any, extra memory components would an agent need, in order to be able to handle incoming ACHIEVE messages. Assume that such messages contain a goal that the sending agent wants the receiving agent to achieve on its behalf. This goal is represented as a belief that the receiving agent should hold at a future time. Also assume that such a future belief cannot always be achieved by just one commitment.

The two parts carry, respectively, 40%, 60% of the marks.

End of paper