Lab Sheet 2 - Dining Philosophers - CITS3213

Complete these exercises and submit answers to these questions.

Make sure you attach a green assignment cover sheet to your answers.

For Q 2.1 - 2.2 and Q 2.5 include the XCircal code you added to the "PhilosophersQuestion.xtc" example file. Indicate whether your code actually runs correctly (partial marks make be given for non-working solutions, no marks will be given for code that claims to work but doesn't).

The XCircal example file "PhilosophersQuestion.xtc" has been provided for this exercise.

Q 2.1 (2 mark)

In the lecture there was a process definition for a philosopher that waited for the fork on his left then right before eating:

```
  Ph <- think takeLeftFork takeRightFork eat dropLeftFork dropRightFork Ph
```

Write an XCircal procedure that takes six events (eating, thinking, and the fork events) and returns a process that behaves like the prototypical philosopher above.

Q 2.2 (2 marks)

Extend the XCircal example sheet "PhilosphersQuestion1.xtc" to construct an instance of the dining philosophers problem, for N=2, using the philosopher behaviour defined in Q 2.1 . Make the Circal System print out the system's behaviour.

Q 2.3 (1 mark)

Draw a structure diagram (in the style given in lecture notes 2, 3 and 4) for the system defined in Q 2.2 (ie 2 philosophers, 2 forks). Label all lines that connect two ports with the associated action name.

Q 2.4 (2 marks)

With the philosopher behaviour given above, the dining philosophers system can deadlock. Describe a sequence of events that can lead to a deadlock. Explain why neither philosopher can perform any action.

Q 2.5 (1 mark)

Repeat Q 2.1 and 2.2, but change the behaviour of the philosopher so that he grabs both forks simultaneously. HINT: consult the lecture notes and provided Circal System notes for the XCircal syntax for simultaneous events.

Q 2.6 (2 marks)

Can the solution in 2.5 deadlock ? Justify your answer using the displayed system behaviour.

Total: 10 marks

Bonus Questions (no marks)

Is the deadlock present in question 2.4 a problem just for N=2 ?

Does your solution in Q 2.5 work for N=3 ?

Can your solution in Q 2.5 "livelock" ? That is, can there be continual movement of forks without eating and thinking ?

DUE: WEDNESDAY 16TH APRIL 2008..please ensure you place a completed green cover sheet on the front of your assignment.