Exercise Solutions
−Chapter 4−

Exercise 4.1.Specify the algebraic components of the event graph in Fig. 4.5.

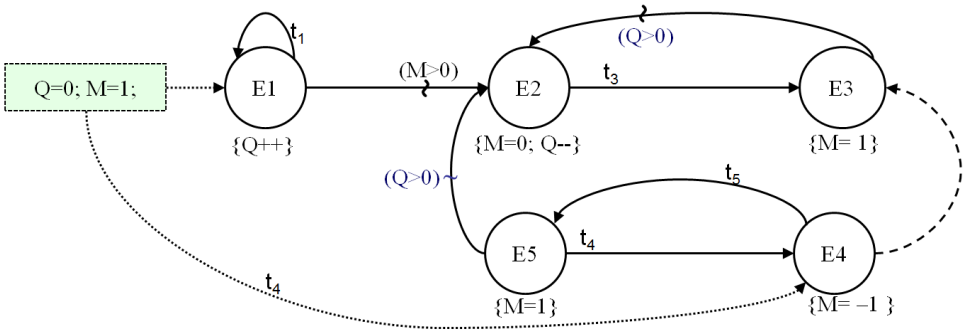


Fig.4.5 Event graph model with an initialization box

<Solution>

$M_{\text{Fig.4.5}}$ = <*V*, *E*, *S*, *F*, *C*, *D*, *A*>, where
V = {v_0 = Initialize, v_1 = E1, v_2 = E2, v_3 = E3, v_4 = E4, v_5 = E5},
E = {$e_1 = (v_1, v_2)$, $e_2=(v_2,v_3)$, $e_3=(v_3,v_2)$, $e_4=(v_4, v_5)$, $e_5=(v_4, v_3)$,
$e_6=(v_5,v_4)$, $e_7=(v_5, v_2)$, $e_8=(v_0, v_1)$, $e_9=(v_0, v_4)$},
S = {Q, M},
F = {f_1: Q++, f_2: M=0; Q--, f_3: M=1, f_4: M=-1, f_5: M=1},
C = {c_1: (M>0), c_2: True, c_3: (Q>0), c_4: True, c_5: True,
c_6: True, c_7: (Q>0), c_8: True, c_9: True},
D = {$d_1 = 0$, $d_2 = t_3$, $d_3 = 0$, $d_4 = t_5$, $d_5 = 0$, $d_6 = t_4$, $d_7 = 0$, $d_8 = 0$, $d_9 = t_4$},
A = {$a_1=a_2=a_3=a_4=a_6=a_7=a_8=a_9$=scheduling, a_5=canceling}.

Here, the initialize box (Q=0; M=1) is regarded as an event vertex.

Exercise 4.2.Simplify the baseline model ofFig. 4.1(b) by removing the Load event.

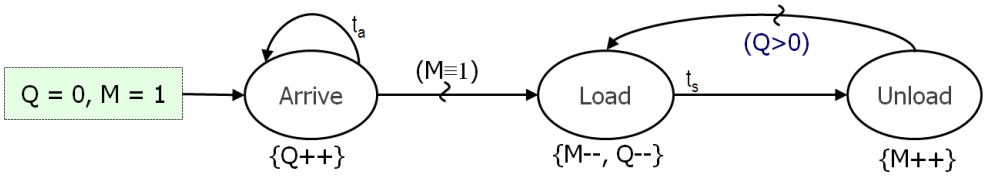
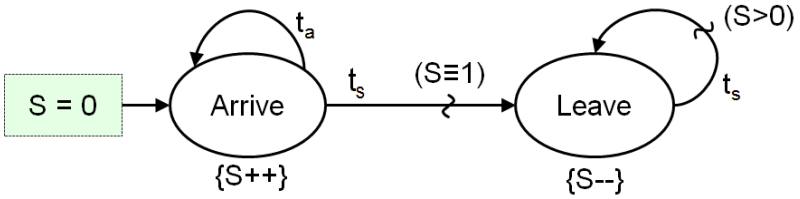


Fig. 4.1(b) Event graph of the single server system (reproduced from Fig.2.1)

<Solution>

If the Load event is removed, the state variable M becomes undefined (i.e., it has to be removed) and the state variable Q is replaced with a state variable S denoting the

number of jobs in the system (jobs waiting and the ones being processed).



Exercise 4.3. Construct a singleserver system event graph for reneging without balking (by modifying Fig. 4.8).

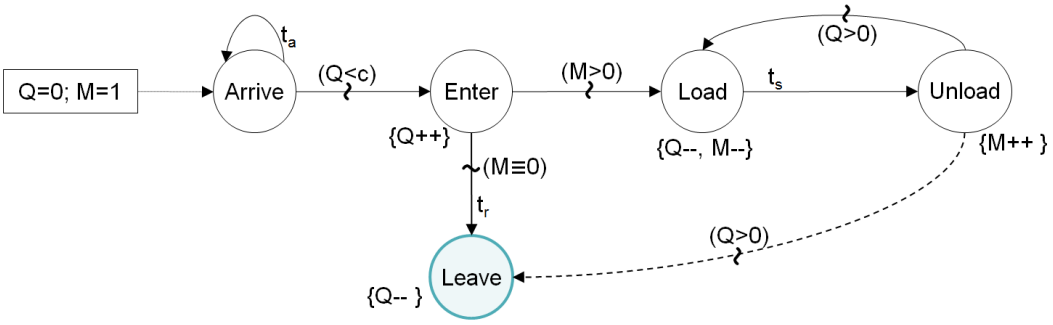
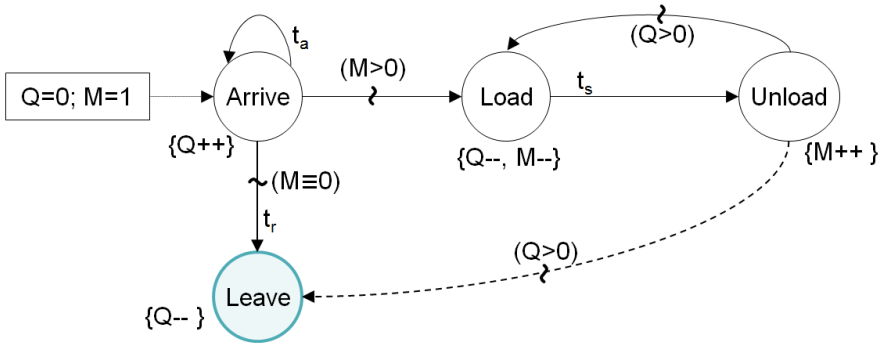


Fig.4.8 Event graph model for reneging with balking

<Solution>

The arc condition from Arrive event to Enter event is removed. Then the event graph model is simplified by merging Arrive event and Enter event into one.



Exercise 4.4. Revise the event graph presented in Fig. 4.10 to make it a partial batched service model.

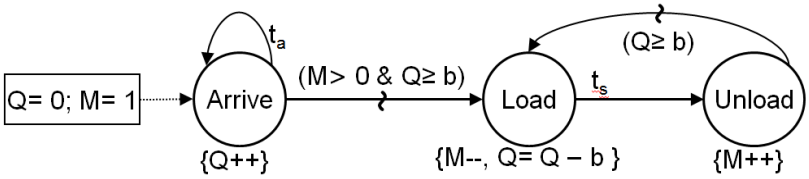
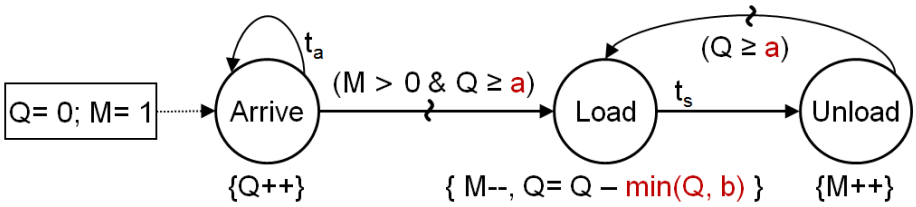


Fig.4.10Full batched service event graph model

<Solution>

The *arc conditions*of the Arrive-Load arcand Unload-Load arc are modified to reflectthe 'minimum number of jobs'constraint($Q \geq a$). Also, the state update at theLoad event is changed such that Qis decreased by ($\min(Q, b)$).



Exercise 4.5. Modify the event graph of Fig. 4.11 to assemble three type-1 parts and four type-2 parts.

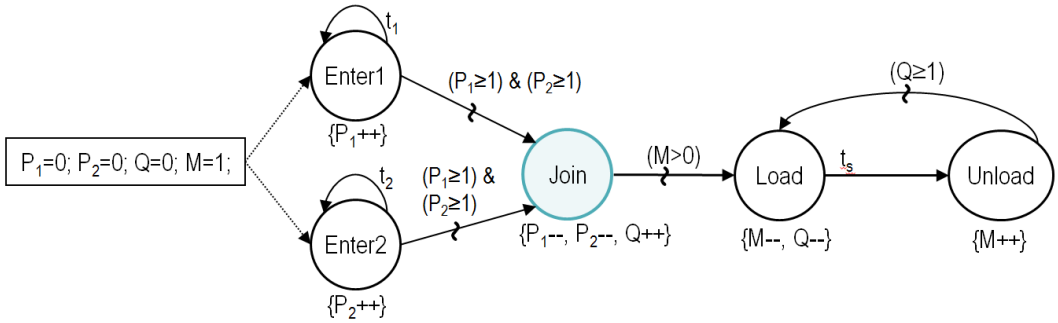
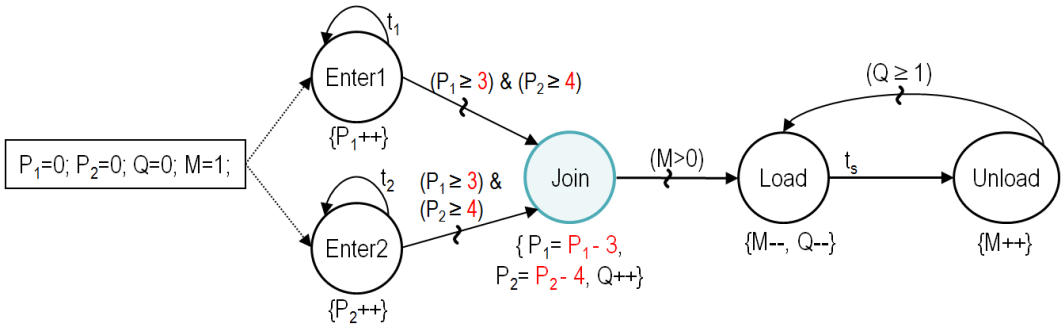


Fig.4.11 Event graph model for an assembly operation

<Solution>

The *arc conditions* of the Enter1-Join arc and Enter2-Join arc are modified from ($P_1 \geq 1$) & ($P_2 \geq 1$)to ($P_1 \geq 3$) & ($P_2 \geq 4$). Also, the state updates at the Join event are changed as P_1-- → $P_1 = P_1 - 3$ and P_2-- → $P_2 = P_2 - 4$



Exercise 4.6. Modify the event graph of Fig. 4.12 to make it a queue length balanced line.

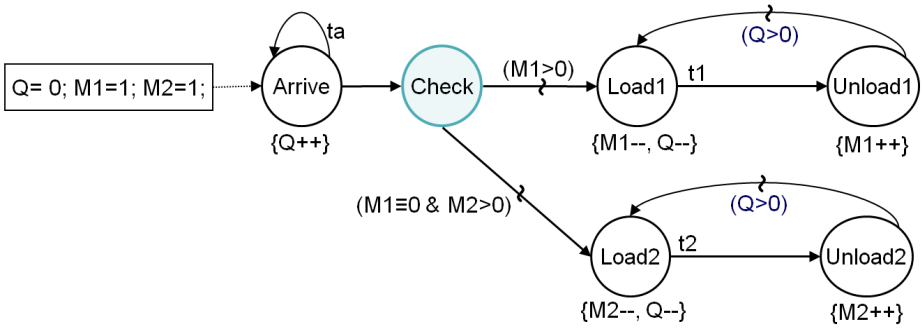
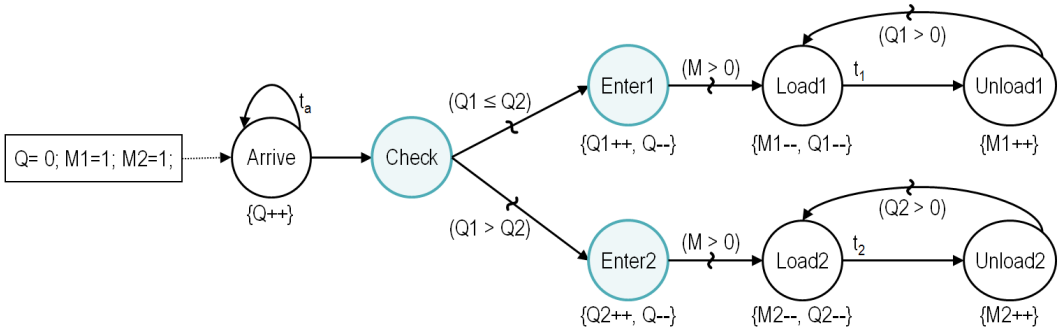


Fig.4.12 Event graph model of a two server system with resource priority

<Solution>



Exercise 4.7. Modify the event graph of Fig. 4.13 so that the job that was interrupted by the failure is reprocessed.

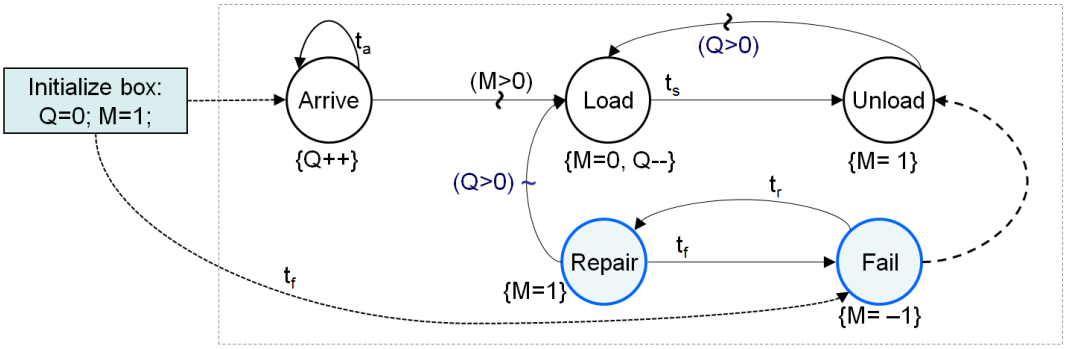
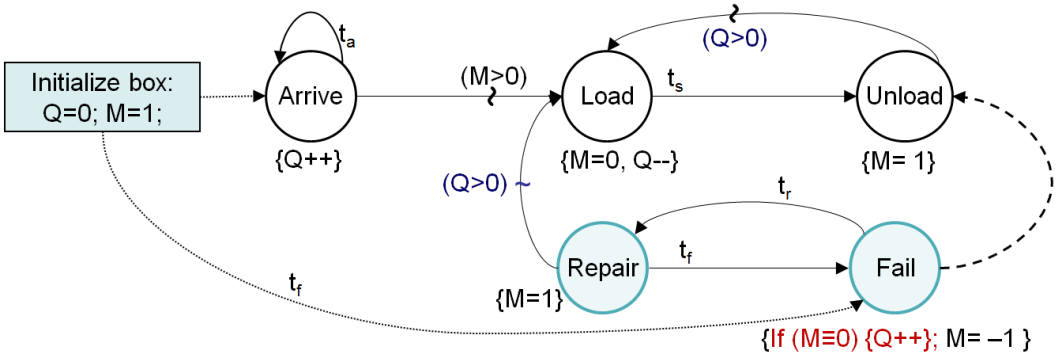


Fig.4.13Event graph of singleserver system with failure

<Solution>

The state change at the Fail event is modified such that Q is increased by one (Q++) if the machine fails when it is busy (i.e. it has a job being processed).



Exercise 4.8. Modify the event graph of Fig. 4.18 so that the time-out jobs are reprocessed.

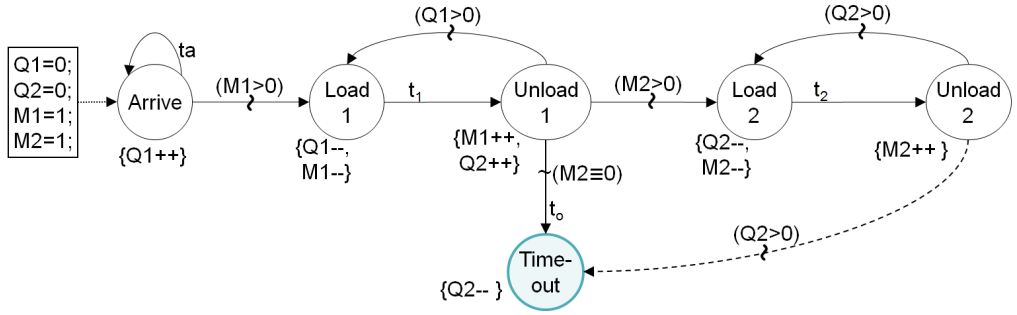
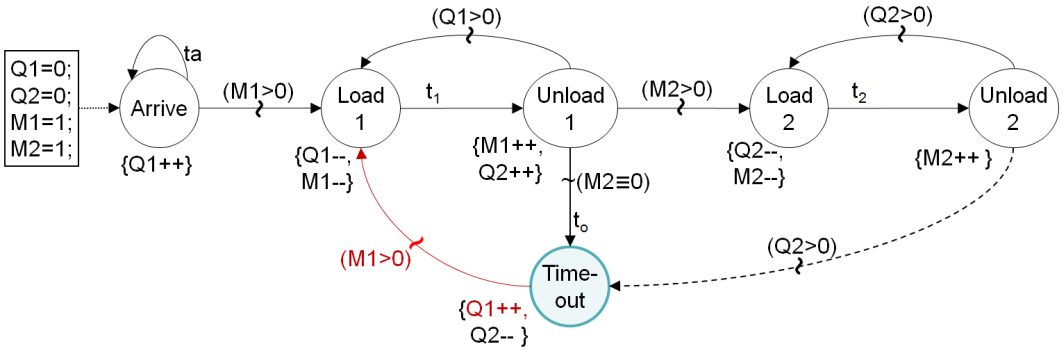


Fig.4.18 Event graph model for time constrained processing with a discarding policy

<Solution>



Exercise 4.9. Build an event graph for the reference model of Fig. 4.20(b).

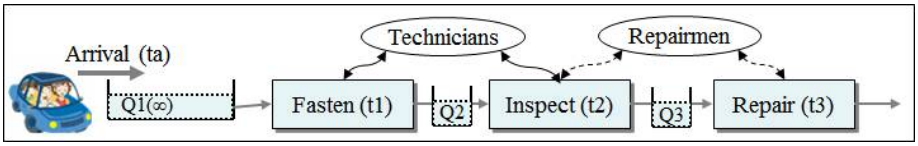
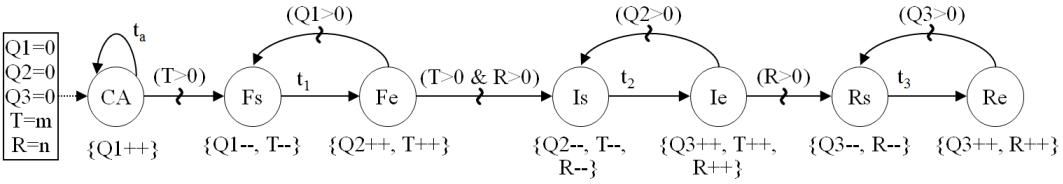


Fig.4.20(b) Reference model of carrepair shop without a*sameoperator* policy

<Solution>



Exercise 4.10. Simplify the PERT event graph of Fig. 4.23 by removing the state variables $\{n_j\}$ whose initial value is 1.

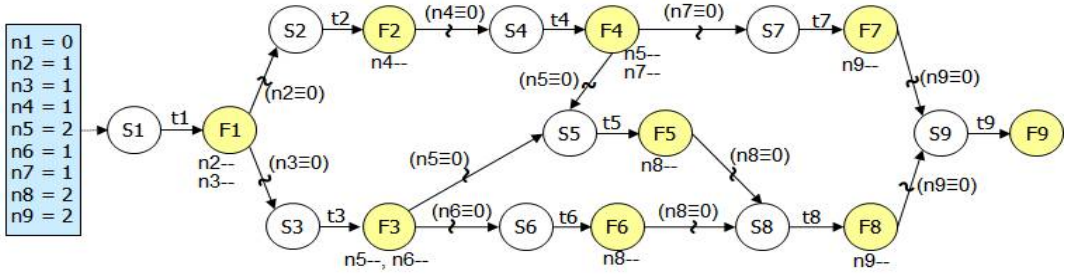
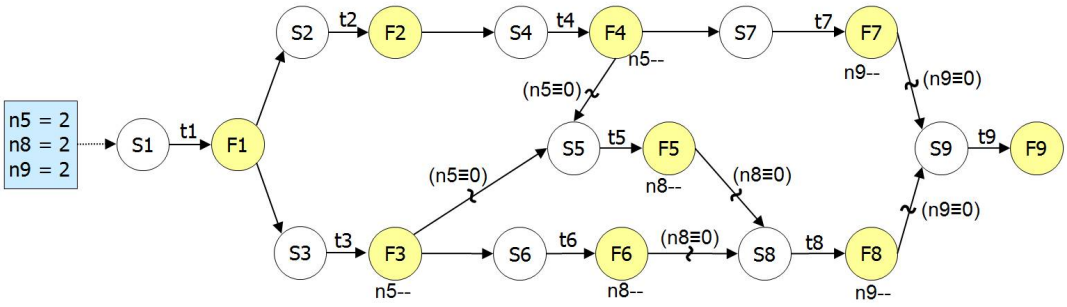


Fig.4.23 Event graph of the PERT diagram of Figure 4.22without resource constraints

<Solution>



Exercise 4.11. Simplify the inline cell event graph model in Fig. 4.34 by removing the FGU event vertex.

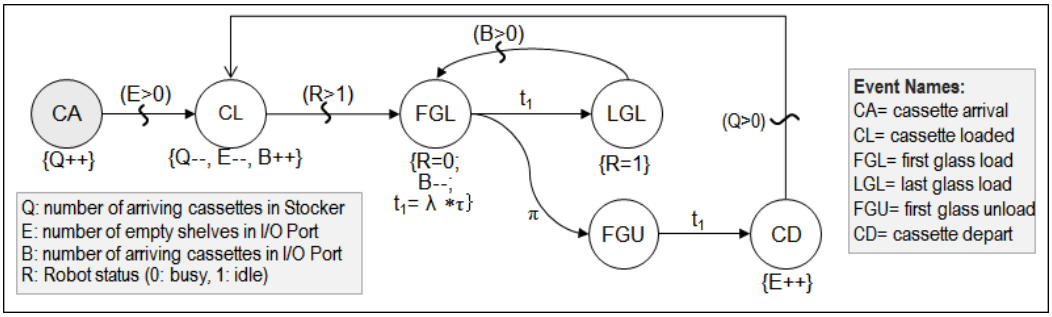
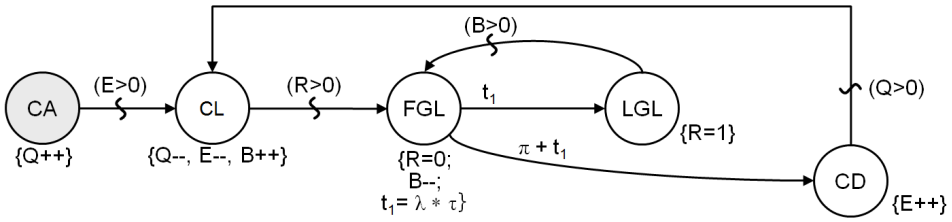


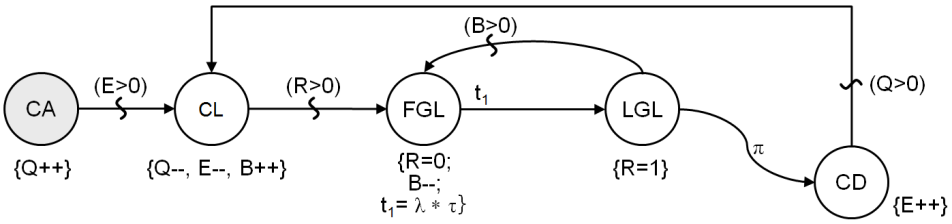
Fig.4.34 Combined event graph model of the inline cell

<Solution>

Case (1):



Case (2):



Exercise 4.12. Write two event routines Execute-Load-event-routine () and Execute-Unload-event-routine ().

<Solution>

Execute-Load-event-routine (Now) {

SumQ = SumQ + Q*(Now – Before); Before = Now; M = M + 1; Q = Q - 1;

Schedule-event (Unload, Now+ Uni(4,6)); }

Execute-Unload-event-routine (Now) {

M = M + 1;

If (Q>0) Schedule-event (Load, Now); }

Exercise 4.13. Modify the event graph in Fig. 4.60 to collect both AQL and AWT statistics.

<Solution>