

Notes sketching an airport system

The handwriting is of Kristen Nygaard

```

    (1)
    SIMULATION class COUNTERACT;
    begin integer index; real sumperiod;

    class station;
    initial: ref (clerk) procedure kind; procedure clerkparam;
    real procedure servicetime;
    begin ref (clerk) server;
    procedure clerkparam; end station;

    station class store;
    begin ref (head) q; q := new head; end;

    station class singleq (m clerks); integer mclerks;
    begin ref (head) q, office;
    ref (clerk) procedure kind;
    kind := new commonq clerk (this station, q);
    q := new head; office := new head;
    for index := 1 step 1 until m clerks do
    begin server := kind; server :=
    activate server; server := new; end;
    end;

    station class multiq (channels); integer channels;
    begin integer mserver;
    ref (head) array q [1: channels];
    ref (clerk) array c [1: channels];
    integer procedure multiq;
    begin
    ref (clerk) procedure kind;
    kind := new clerk (this station, q [index]);
    for index := 1 step 1 until channels do
    begin q [index] := new head; server := kind;
    clerkparam := c [index]; activate server; server := new; end;
    end;

    process class clerk (location, q); ref (station) location; ref (head) q;
    initial: procedure wait; ref (customer) procedure choice;
    real procedure servicetime;
    begin ref (customer) selected; real a, b;
    procedure wait; passivate;
    ref (customer) procedure choice; choice := q.first;
    real procedure servicetime; servicetime :=
    location.servicetime;
    repeat: if q.empty then passivate wait;
    selected := choice; selected.out; cancel (selected);
    location.server := new clerk;
    hold (servicetime); activate selected after cancel;
    selected := new; go to repeat;
    end clerk;

    (2)
    class clerk class commonq clerk;
    begin procedure wait;
    begin into (location) sumperiod, office;
    passivate; out end;
    end commonq clerk;

    process class customer;
    begin real servicetime; depletion; integer mserver;
    procedure into (location); ref (station) location;
    begin ref (head) qselect; ref (clerk) cselect;
    inspect location when singleq do
    when singleq do
    begin qselect := q; cselect := office.first end;
    when multiq do
    begin qselect := mserver; cselect := q [mserver];
    when store do qselect := q;
    this process into (qselect); activate cselect after amount;
    end & into;

    procedure maxtime (T); real T;
    begin deactivate amount at T; if T hold then out end;
    end customer;
  
```

process class arrival;

```

    begin
    repeat: inner; if time < sumperiod then go to repeat;
    end arrival;
    end COUNTERACT;
  
```

(B) COUNTERACT begin

```

    ref (station)
    ref (station) checkin 1, checkin 2, bank, kiosk, control,
    dom lobby, int lobby;

    arrival class charter;
    begin hold (msgap (3));
    activate new charpass; end;

    arrival class international;
    begin hold (msgap (2));
    activate new intpass; end;

    arrival class domestic;
    begin hold (msgap (2));
    activate new dompass; end;

    multiq class special norm counter;
    begin procedure clerkparam;
    begin server.a := INPUT; server.b := INPUT end;
    real procedure servicetime;
    servicetime := normal (server.a, server.b);
    end special norm counter;
  
```

(C) multiq class speccounter;

```

    begin procedure
    real procedure servicetime;
    servicetime := server.selected.servicetime;
    end speccounter;

    singleq class charcounter;
    begin real a, b; a := INPUT; b := INPUT;
    procedure clerkparam;
    kind := new charpass;
    real procedure servicetime;
    servicetime := normal (a, b);
    end charcounter;

    singleq class kioskcounter;
    begin
    real procedure servicetime;
    servicetime := server.selected.servicetime;
    end kioskcounter;

    customer class charpass;
    begin into (checkin 1); passivate;
    deactivate depletion := time + normal (30, 5);
    into (checkin 2); passivate;
    if (depletion - time > 10) & dur (0.2) then
    begin into (bank); maxtime (depletion - 3) end;
    into (control); passivate;
    into (int lobby); passivate;
    end;
  
```

(A) AIRPORT DEPARTURE