Carleton University
Department of Systems and Computer Engineering

SYSC-3120 **Software Requirements Engineering** **Winter 2015**

Laboratory 9

Attendance/Demo

- To receive credit you have to both show you solution to the TA and submit your work to cuLearn.
- For those who don't finish early, the TA will ask you to show whatever you have completed, starting at about 15 minutes before the end of the lab period.

Instructions

To better benefit from the work being asked, it is recommended to work in groups of two or three (not more). Your answer document should contain the names of all the group members.

Exercise 1

The purpose of this exercise is to model the start/run sequence for the engine of an aircraft by using two communicating state machines. One state machine will model the power source controller while the other will model the engine controller.

An engine has three possible electrical power sources:

- An engine alternator, which is only available when the engine is running. The alternator produces electricity from the running engine itself;
- An airframe auxiliary power unit (APU), which must be used during starting;
- An airframe battery, which is only sufficient to power electronic systems and to start the APU.

The start/run sequence of an engine is the following:

- When the engine is not running it is either stood or being started;
- To start the engine, the power source must first be switched from battery to APU (which is itself powered by the battery), which starts the electrical starter motor. This results in the engine being started until "light-off" is achieved;
- If light-off is not achieved after a period of 60 seconds, the engine must be purged to remove un-burnt fuel before it is returned to the stood state. At the same time, the power source must return to "battery";
- If light-off is achieved, the engine reaches ground idle and can be accelerated to flight idle. The engine is considered running in both these (idle) situations. When this happens, the power source switches to "alternator";
- The engine must be returned to ground idle before it is shut off;
- To avoid damaging the alternator, power should be switched back to battery before shutting off the engine.

You will assume the engine controller and the power source controller are modeled by two state machines that communicate through signal events.

For the purpose of modeling, you will assume the following:

- The power source controller is referred to by a reference called psc;
- The engine controller is referred to by a reference called ec;
- The current state of the power source controller is modeled by an attribute, called pscState, which values are from an enumeration of the different states the controller may be in;
- The current state of the engine controller is modeled by an attribute, called ecState, which values are from an enumeration of the different states the controller may be in;
- The condition to light-off, which realistically would be defined in terms of torque, pressure, temperature values, is evaluated by a (query) operation called lightOff(). A query operation only computes something and/or returns a value without changing the state of the system. In this case, lightOff() returns true if and only if light-off has been reached.
- The engine controller can receive the following signal events:
    - starterOn, which indicates that the electrical starter motor has started to start the engine;
    - toFlightIdle, which indicates acceleration of the engine;
    - toGroundIdle, which indicates a return to the ground idle state;
    - turnOff, to completely turn off the engine. Only to be done while in the stood state;
    - engineShutOff, to shut off the engine, i.e., to return to stood.

- The power source controller can receive the following signal events:
  - switchToAPU, to switch the power source to the APU;
  - switchToAlternator, to switch the power source to the Alternator;
  - switchToBattery, to switch the power source to the Battery;
  - turnoff, to completely turn off the power source. Only to be done while on battery.

Exercise 2

The behavior of the instances of a class is represented with the following state machine. Note that S1 is a composite state with three concurrent substates S2, S6 and S9).

Once created, an instance of that class receives the following sequence of messages. What is the list of states the instance goes through? What are the actions performed and the events sent?

Sequence of received events: `ev1`, `ev2`, `ev6`, `ev4`, `ev2`, `ev5`, `ev10`, `ev8`, `ev11`, `ev4`, `ev6`, `ev3`, `ev12`

For this question we make the following assumptions:

- all the receptions (events), actions, and sending of signals are instantaneous;
- x is big enough to prevent the timeout transition (after(x)) to be taken;
- events are sent one by one: when a transition is fired because an event has arrived, no new event arrives until the state of the state machine has stabilized (until all the states are reached and actions/activities completed).