CS 578 – Software Architectures
Spring 2010
Homework Assignment #3
Due: *Thursday, April 1st, 2010*

In this assignment, we provide you with the domain model for the robotic scenario in assignment #2 as well as a simulation and graphical interface which simulates and presents the robots moving and following each other. You are expected to provide the C2SADEL specification for the robotic scenario and use a software architecture modeling and analysis tool, called DRADEL, to check the syntactic correctness as well as structural and behavioral soundness of your provided C2SADEL specification you developed. You are expected to provide a C2SADEL specification which enables

1. parsing (i.e., syntactic checking),

2. checking structural constraints,

3. type checking (i.e. behavioral checking), and

4. code generation

in DRADEL. Your C2SADEL specification must be able to "pass" all four of these steps. For example, if you are not able to pass the structural constraint checking or type checking steps, you will not get credit for your implementation. The generated code skeletons are compilable on top of Prism-Lite, a simple architectural middleware platform written in Java, but they do not provide any application logic (also known as business logic), which in your C2SADEL specifications is reflected in the pre- and post-conditions. You are expected to implement the application-specific logic for the robotic scenario, by "filling in the blanks" of the automatically generated Prism-Lite code. The application must run with five robots, and the convoy should not break if the leader robot is out of battery.

Domain Model

Figure 1 depicts a domain model for the robotic scenario in assignment #2. The domain model in figure 2 provides the domain model for a single robot which is applicable for all the robots including the leader robot. In this assignment all the robots, including the leader robot may fail due to battery depletion. However, the robots convoy should not break in any case. When the leader robot is low in battery the front most robot should become the leader and when any follower robot is about to die, the robot behind it should switch to follow the robot in front of it. The convoy should start with five robots and should continue working until all the robots batteries are depleted.

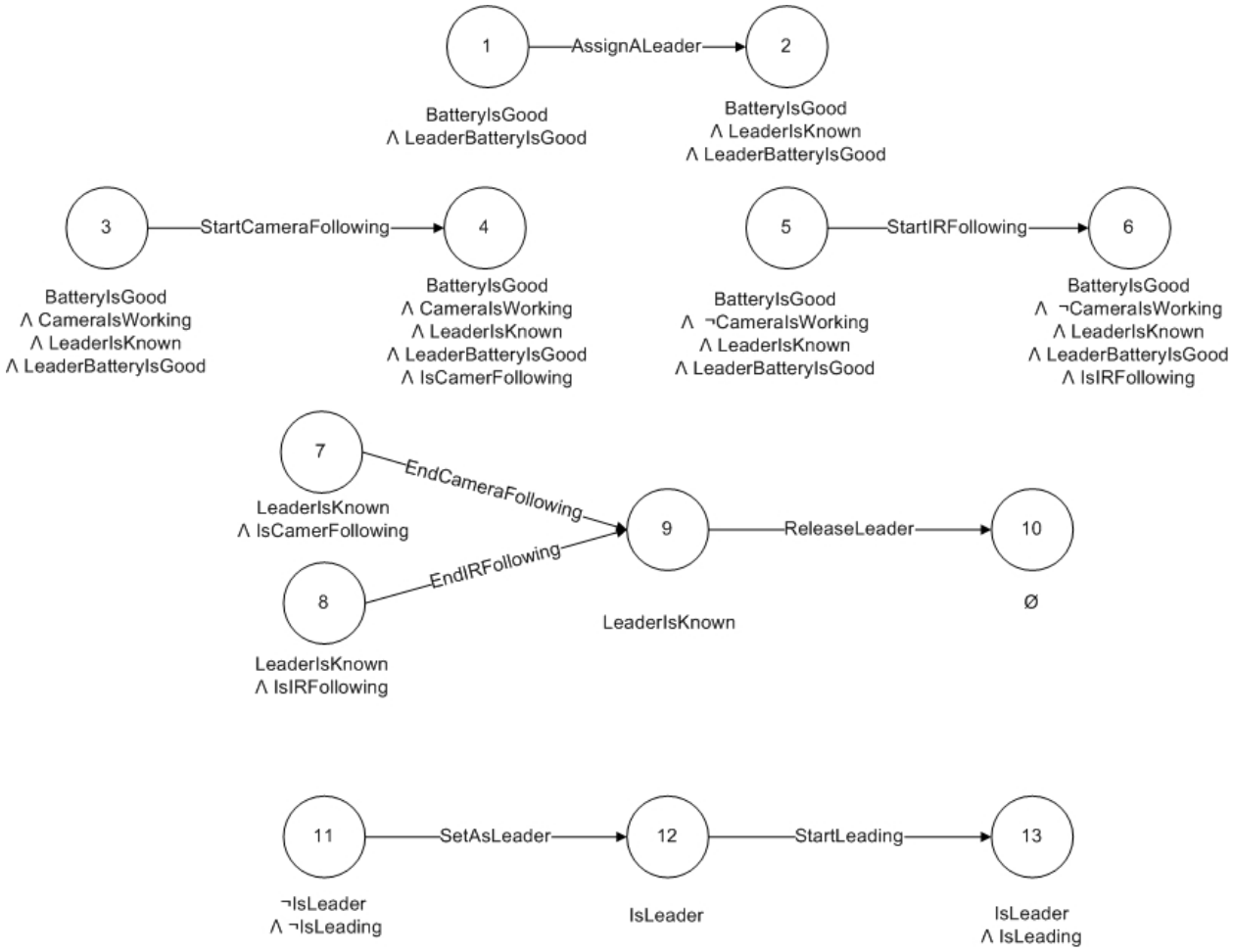

Figure 1

Simulation Environment

We provided two java classes to simulate and represent the environment and robots: *Robot* and *Environment*. The *Robot* class simulates a robot that has a color and an IR code and that can follow another robot using its camera or IR sensor. The camera of a robot might be broken at the beginning or it might break later. The battery of the robot depletes as time passes. The battery depletes at the highest rate when the robot is camera following another robot and is at the lowest rate when it is not moving. When the robot is leading the convoy or performing the IR following the battery depletion is at a medium rate. Note the robot functions become unavailable when the battery level is at 0 percent. On the other hand, the *Environment* class contains the *Robot* and associated classes and shows the graphical interface. When a robot's battery is dead it will be deleted from the environment.

Figure 2 shows a snap shot of the graphical interface of the *Environment* class. The boxes on the top row show the robots which have some battery left. The Bottom row shows the current convoy of the robots. This picture shows the condition in which the red robot is the leader, the blue robot is color following the red robot and the orange robot is IR following the blue robot, and two of the robots are already dead.

As part of the assignment, you are provided an Example C2 architecture that describes how a robot is added to the environment and how the components should interact with the robot.

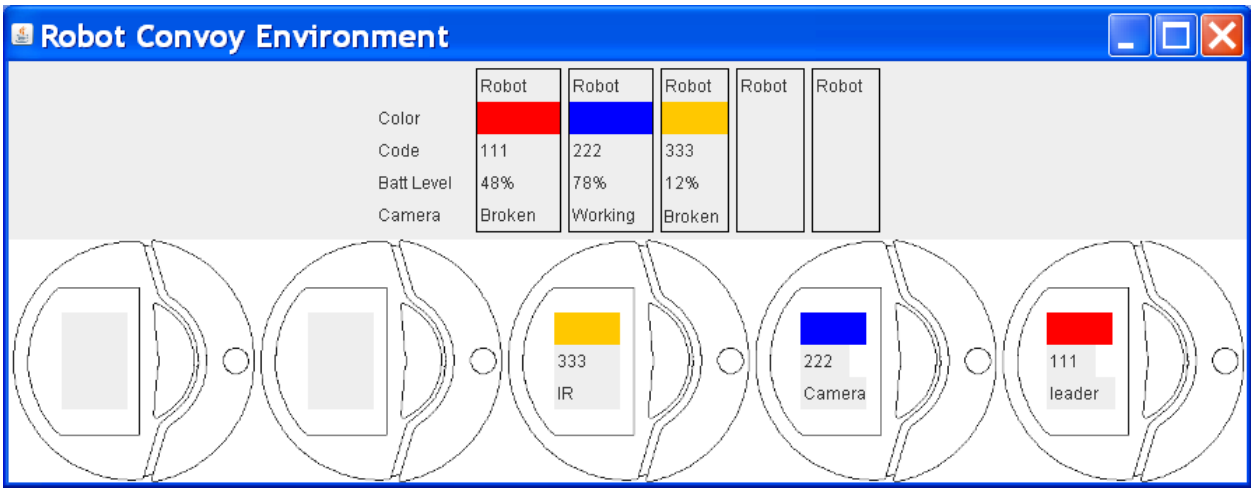

Figure 2

The files required to complete the assignment are located at

http://csse.usc.edu/classes/cs578_2010b/HW3_Package.tar.gz

This package contains three directories: DRADEL, Prism-Lite and the robotic example. It also includes a jar file containing the class files related to the simulation environment. Read the available documentation carefully as questions and emails regarding material that you should have already covered will go to the bottom of the priority list. After returning from Spring Break, depending on time, availability, and where we are in the lecture, we may provide a demo of Prism-Lite in class to further inform your effort.

Deliverables

You are required to turn in a single tarred and gzipped file that contains

1. Your C2SADEL specification,

2. The implementation files,

5. A log file which captures the state of the convoy whenever it is changed.

Your C2SADEL files must be packaged such that they can be incorporated into DRADEL without any changes. Your Java files must be packaged such that they can be deployed on Prism-Lite and executed without any changes.

Below, we have provided the expected format for the log file (the log file should be generated each time that your application is run):

```
1:20:23.245	Red(leader)
1:20:23.430	Red(leader)	Blue(Camera)
1:20:23.732	Red(leader)	Blue(Camera)	Green(Camera)
1:20:24.121	Red(leader)	Blue(Camera)	Green(Camera)	Orange(Camera)
1:20:24.522	Red(leader)	Blue(Camera)	Green(Camera)	Orange(Camera)	Yellow(IR)
1:21:02.010	Green(BatteryDepleted)
1:21:02.210	Red(leader)	Blue(Camera)	Orange(Camera)	Yellow(IR)
1:21:53.107	Orange(CameraFailed)
1:21:54.210	Red(leader)	Blue(Camera)	Orange(IR)	Yellow(IR)
1:22:30.107	Yellow(BatteryDepleted)
1:23:30.410	Red(leader)	Blue(Camera)	Orange(IR)
```

Note that each row in the log file contains columns which specify:

- A date/time in the simulation.
- The robot's color (and in parenthesis its current state, one of leader, camera , or IR) for each robot in the convoy, leader ongoing.

Note that each column in the log file is separated by a single tab. Also note that you should log camera and battery failure events in each robot, on a per line basis, as shown above. On those lines, you need only to log:

- A date/time in the simulation.
- The robot's color (and in parenthesis, what failure occurred).