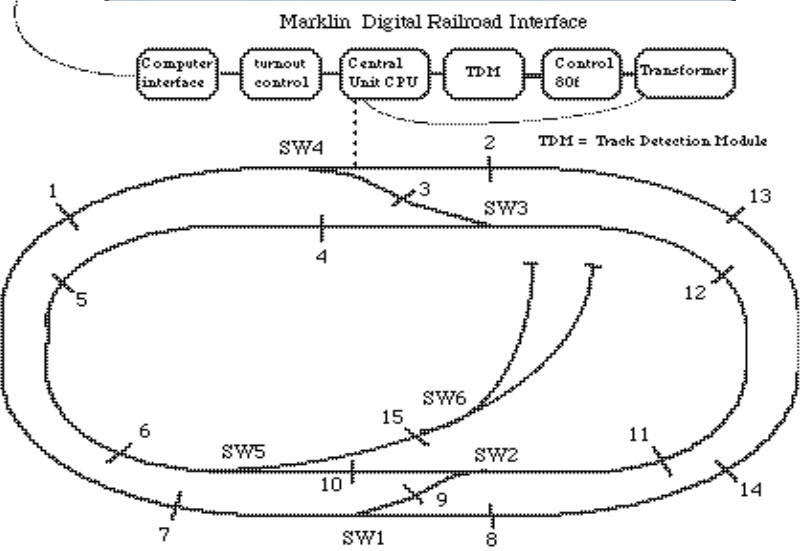
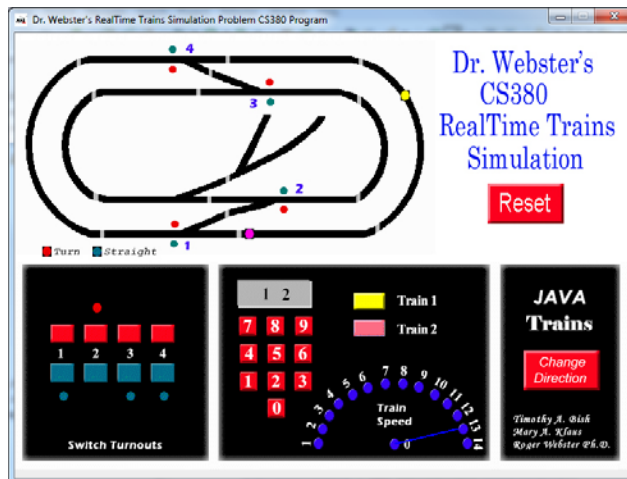


CS380 Operating Systems
Real-Time Concurrent Simulation Programming Assignment
Laboratory Assignment
 Roger W. Webster, Ph.D.
 Intelligent Machines Laboratory
 Department of Computer Science



The objective of this assignment is to gain experience in the following areas:

- Programming concurrent tasks/processes.
- Sending and receiving messages.
- Java Thread and Mutex programming.
- Operating System programming problems.
- Simulating real-time programming environments.

Write a concurrent, message passing program that performs a simulation of a real-time information passing among processes. Create a Java thread which will simulate reading the *reed* switches on the real-time digital model railroad system and pass the info to the controlling thread. The reed contacts will trip when an engine crosses the contact on the track. The GUI thread should be capable of accepting commands from the user :

- start train1,
- start train2,
- stop train1,
- stop train2,
- slow down & speed up each train,
- throw the two turn-out switches (straight and curved)

You should create separate threads for each train and make it go around the track. Make sure your controller does not allow a train crash. Do not busy wait, and make sure mutex is guaranteed on switches and train collisions. See diagram in handout for details, I have built the graphics for the GUI, see the helper code is on my web site.

Some Notes:

When your controller sees there is an urgent/imminent collision, then it is to set the rear train to stop. This must not Busy Wait, i.e., setting the speed to 0 should not busy wait.

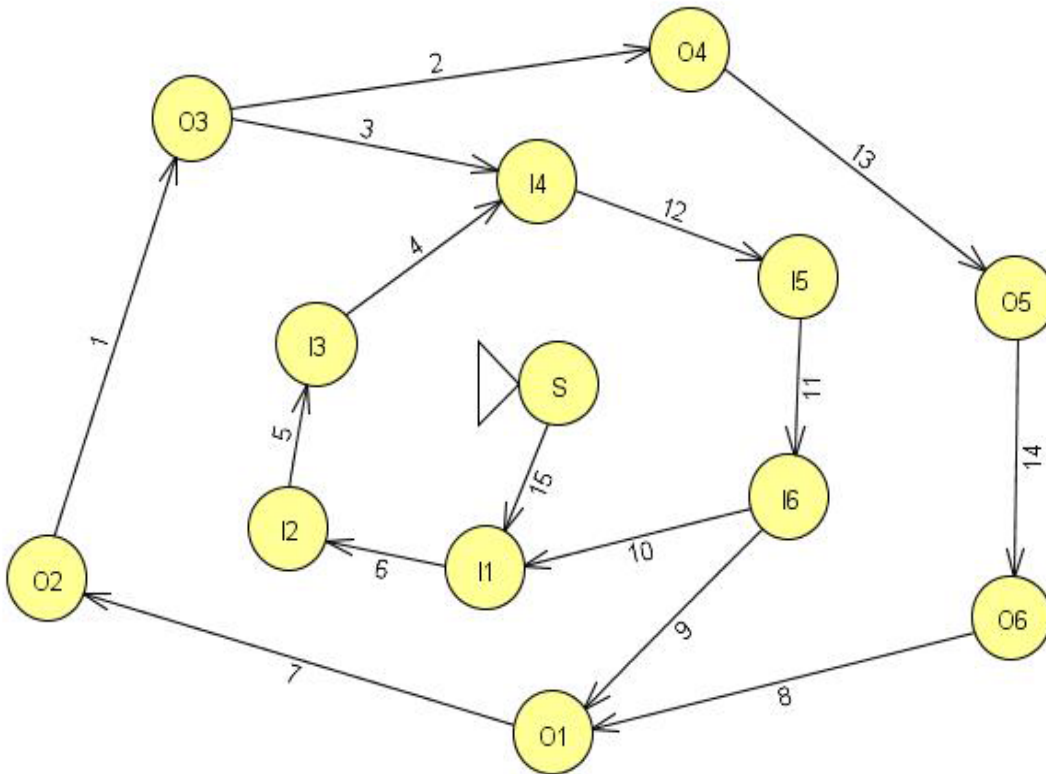
You can use any and all of the tools of Operating Systems to accomplish these things:

Semaphores, wait and notify, synchronized methods. You can not write this controller without one of many of the above and not busy wait.

Also, the GUI can simply ignore the switching if there are trains that will collide or be unstable because of it.

Other Notes:

- (1) Have the GUI send to the trains the DesiredSpeed that the user wants the train to run at.
- (2) Store the DesiredSpeed in each train.
- (3) When there is no collision imminent, then set the train's speed to the DesiredSpeed as set by the user at the GUI.
- (4) Any collisions, always set the rear train to slow down or stop, do not speed up the trainAhead. A train stop or speed of 0 is NOT to Busy Wait!!
- (5) For AutoRun mode then you completely control all speeds regardless of User at the GUI.



Good Luck!