Homework 3
Analysis of Software Artifacts
Due Date: Feb 15, 2000 (Tuesday)

Note: For this homework, you only have one question. We gave only one question, because it will take you considerable amount of time to change the SMV specifications according to the directions given below. As in the earlier homework, before you start changing the SMV model, write a *design document*. Outline your design strategy in the design document. Please be as detailed as possible. Show the design document to a fellow student, i.e., the *reviewer*. Mention the reviewer's name on the document. Keep the design document *consistent* with your specification. Please submit the *modified SMV specification AND the design document* with the homework. This will also help the TA grade your homework and will help you with the homework.

Question 1 (Modeling 100points): Use the SMV specification that TA will provide as a solution for Homework 2. You will enhance the specification in the following way:

(Part A):
Multiple Priorities
Users have two levels of priority: *high* and *low*. Assume that we have two users of each priority. Within users of the same priority use the round-robin scheduling policy[1]. If two users of different priorities are waiting, the user with the higher priority gets the machine.

(Part B):
More coins
Assume that each drink costs *fifty cents*. User deposits *dimes* or *quarters*. As soon as the user (who has the vending machine) has deposited coins worth greater than or equal to fifty cents, they can get the drink. Of course the vending machine will have to return the spare change.
(`Hints:`) You will probably have to keep an extra variable called `amount-accrued` which keeps track of how much the user has deposited. In the vending machine, there should be a variable called `amount-returned` which is equal to

[1] We will discuss round-robin scheduling in the lectures

the spare change. First you should figure out how many possible values could the variables `amount-accrued` and `amount-returned` take. Do not use arithmetic in `SMV` to keep track of `amount-accrued` and `amount-returned`. Use explicit transitions to simulate addition, e.g., if `amount-accrued` is `twenty` and the user deposits a dime, then `amount-accrued` becomes `thirty` in the next step.