

Project: SM Travel

Final Deliverable - Comments

1) Problem Description: Good

- SUI Details: Introduction: Our study focuses on the US site.

2) Project Goals:

- Parameter RG.TrunkLine.numReserved: I am wondering on the maximum number of reserved lines. Given that regular customers call less frequently than gold or silver customers (the arrival data models show about the same arrival rates), would it be possible that the number of reserved lines could occupy half the truck lines? In fact in your experimentation, you use the value 100 instead of 50.
- Experimentation:
 - In the intro, should mention also finding the number of reserved lines, not only the number of trunk lines.
 - Step 5: ii and iii are the same. I suspect you want to remove one of these. Also b and c are the same; I suspect c should be regular customers instead of silver customers.
 - The presentation has changed since the original. I am assuming the following: when I see run and experiment, this means that you go to step 4b after that step. I expect that at the end of 5-a-iv you would also run an experiment. Also if the gold wait-times are satisfied, then you go to step b or do you check b in anycase (4a says add an operator, so I am assuming that steps 5b and 5c are only evaluated if in previous steps wait times are satisfied).
 - How did you arrive at the logic for the selection of operators? Why base it on results of each shift? You may know why, but it is not evident from your presentation.
- Output: I see no output variables that give the satisfaction level at the end of each shift. Is this missing?

3) High Level CM:

- Structural View:
 - First note: “ordering will be based on the time of the call”. Is this really true? Given that Calls in the gold line will be serviced first. In fact, should this note be more about dividing the call queue (as presented in the SUI details) into three queues. Should place the notes after introducing the entity categories. The second note could be incorporated into the RG.Operator category description.
- Behavioral view
 - Call Life Cycle: Could make dashed line more evident and have dashes extend all the way to the Service activity.
 - StaffChange: customer should be call or any life-cycle.
 - EstimateWaitTime: Should describe the interruption.

4) Detailed CM

- Constants:
 - Why defined the STAFF_CHANGE_TIMESEQ in the constants table? It should be part of the StaffChange action. What is the value -1 mean in the context of ABCmod? This is an implementation issue that need not be reflected in the CM. Remember that the SM reflects the CM, not the reverse. In fact I discovered that this constant is not used anywhere in the SM (i.e. the TimeSequence of the StaffChange was properly implemented in the StaffChange class without it).
 - NONE – what does “null value” mean in the context of the CM.

- I see that you have added constants required for the SM in the CM – this is NOT required, nor desired. The SM reflects the CM not the reverse.
- Call Entity: waitTime is not present in the SM, nor should it be present in the CM.
- Initialise: Should the queues be initialised and the number of RG.Operator entities? Even if in Java by default on creation they are initialised properly, in the CM, their initialisation should be flagged. Remember that ABCmod is language neutral.
- Output: I still do not see how the model will support the Experimentation strategy as there are no output variables that gives satisfaction at the end of each shift.
- StaffChange:
 - TimeSequence: the -1 has not meaning in ABCmod, should be removed (this is an implementation detail).
 - Compare the CM Event SCS to the SM actionEvent() method. Totally different. A UDP was used. Need to be consistent between the CM and SM. IN this case the SM has a much more elegant representation – it should be reflected in the CM.
- RegularCallArrival action
 - Starting Event SCS: missing iC.Call \leftarrow SP.Derive(Call)
 - Should UPD.UpdateNumArrivalsOutput(iC.Call) be called outside the IF instruction, that is for all call arrivals to count the number of calls that have arrived. Otherwise you will only count the calls with busy signals. Also applies to the CardHolderCallArrival.
- EstimateWaitTime activity
 - Interruption precondition: what is Q.CallLines[].spIsEmpty(). The notion of a “method” in ABCmod does not exist. This should be Q.CallLines[iC.Call.uCuType].n == 0 and the implementation in the SM should conform.
 - UDP.CheckAvailbleOperator: Pseudo-code is hard to interpret – it gives the how not the why. 1) For a Gold Customer type, an operator is available if a Gold operator is free (RG. ...) or silver operator is available (RG. ...) or a regular operator is available (RG....). For 2 and 3 it's not necessary to repeat the checks, i.e. simply state, for a SILVER customer and operator is available if either a SILVER or a REGULAR operator is free. This is much simpler to undertand, more compact and contains all details for implementation.
- Service Activity
 - The two UDPs can be combined and return a vector <operatorId, callLineId> (in the SM, return a reference to an array of two values). The IF pseudo-code instructions are not necessary. Instead: Return the operator opId and callId when the service of a call is possible according to the following conditions: Gold operator is available (RG. ...) and the gold queue is not empty (Q.CallLine[Gold].n > 0) OR ... But I am glad to see that you did respect the ABCmod paradigm.
- Design of validation Experimentaion: I suspect this was not changed since the last deliverable. Need to present the program's display to represent the state of the model. What to describe the behavior to verify, not the activity actions (although I see why you did this).

5) Simulation Model:

- Constants class: Add comments to the SM required constants to indicate why they are required in the SM. Also some of these should be part of the CM, such as NONT and STAFF_CHANGE_TIMEREQ, but they should be declared in other classes: e.g. STAFF_CHANGE_TIMEREQ should be part of the StaffChange activity.
- TrunkLines class: whenever adding to the SM, such as the toString() method, always document the changes to have a reader/reviewer understand the addition.

- CallLine class:
 - Instead of “callLine” as the variable for referencing “ArrayList” object, using “list” would be more consistent with the CM.
 - spIsEmpty method: there is no SP.IsEmpty procedure in ABCmod. Should use getN() == 0 instead to be consistent with ABCmod (which surely is expressed as Q.CallLine[id].n == 0).
- UDPs class:
 - Nice to see that you have documented the methods with a header. Good Job!
- Good use of data structures in various UDPs and RVPs – nice and elegant.
- Service Class
 - startingEvent method: Why the casting of the value returned by spRemoveQue method? (I removed it an no error occurred).
- SMTravel class: you do not need to use the stopCondition() method, you have an explicit right hand side of the observation interval.
- Design Section;
 - How does q.CallLine.get(iCCall) serve as reference to a Q.CallLine[id] entity? Again should not use spIsEmpty() within an entity class.
- Validation:
 - Can comment on output of state: Should numBusy attribute be displayed for RG.Operator entities?
 - Only demonstrated the staff change behavior. I see additional output Experiment Case 1, Case 2 etc. But am unsure how to interpret the output. It looks like you are displaying only the output variables.
 - I am a bit surprised at the values for the PercBusy values given the issue I discovered with updating these output variables. I did some investigation and discovered that the UDP was called in the EstimateWaitTime Terminating event. This compensates for the confusing logic in the
 - I cannot agree with the report on Verification and Validation – you cannot only look at output results, you must look at changes in the state of the model.

6) Experimentation and Output Analysis

- Before undertaking actual experimentation for solving the problem, you must determine the appropriate number of runs required for an experiment to obtain good confidence intervals. I do not see this. In your Experiment #1 section, it looks like you are using 10 runs, you should be starting with at least 20.
- I do note that an evaluation on number of runs presented at the end of the section. Given the number of outputs, 0.9 confidence is to low (see discussion on overall confidence). This should be increased to a t least 95% and 98% would probably be a good value.
- I have done a cursory review of your experimentation, and unfortunately don’t have the time to do an in-depth evaluation. I don not know how you could have carried out the strategy expressed in the project goals, given that the model does not output the satisfaction levels at the end of each shift, unless you meant end of the observation interval (this does not give enough information about satisfaction levels at different times of the day).
- The only real output I see is at the end of the report which raises considerable questions. First 10 runs is clearly not sufficient according to the evaluation criteria of 10%. But the output values are very low. I do not understand the difference between the 3 cases, but surely there are cases where the satisfaction criteria is not reached. This leads me to have some doubt on how valid the model truly is.