

MSIM 406/506: Introduction to Distributed Simulation
Homework 4: Conservative Time Management

Purpose:

To design, implement and test a distributed simulation executive utilizing null messages and lookahead to implement conservative time management.

Problem Statement:

You are developing the simulation executive in the absence of a specific application. Once implemented, your simulation executive should work with any application that utilizes the supplied interface. Therefore, you will develop your simulation executive and then a test harness to fully test your product. On completion, your simulation executive will be evaluated based on its ability to work with an instructor developed test harness to which you will not have access.

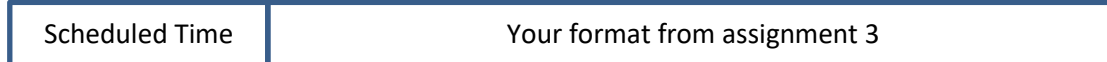
Procedure:

1. Design your simulation executive.
2. Implement your simulation executive. This should include a software architecture identifying your queues and behavioral and functional descriptions for all public methods. Totally hide the fact that you are using MPI within the executive. The executive has methods to get access to relevant information, such as rank, and you may need a static method in the executive to initialize the communication prior to creating an airport.
3. Design a test harness. Recognize that you will need to implement objects that will schedule events remotely and methods that events will invoke.
4. Implement your test harness, then test your simulation executive.

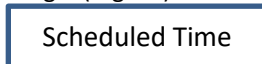
Message Format:

You are to utilize the following message format:

Normal Message: (tag = 1)



Null Message: (tag = 0)



Supplied to you:

1. EventSet.h: A linked list implementation of an event set. You may use it if desired or utilize one you developed in MSIM 331 or 603. If you use your own, make sure it has implemented methods for IsEmpty() and GetTime() where is empty returns true if the event set is empty and GetTime() returns the time of the next event. Neither method modifies the state of the event set.
2. SimExec.h: The declaration for the simulation event class you are developing. The class is virtual, so you should create your own simulation executive class that inherits this class. But this class will be the interface used to access your simulation executive. External to the executive, no other methods may be accessed (any new methods that you add except for constructors/destructors must be private).

Deliverables:

1. Simulation executive design (NO C++ CODE!!!). This should include the architecture of your simulation executive as well as the algorithms for each method.
2. Description of testing. Test harness design (NO C++ CODE!!!).
3. Zipped project directory with:
 - a. Fully documented .h file and .cpp file implementing the simulation executive.
 - b. Test harness.
4. Project status: please provide a description of the status of your effort, i.e. what works and what may not work (hopefully everything works).

Assessment:

1. Simulation executive design	5 pts.
2. Testing/test harness design	5 pts.
3. Implementation	5 pts.
4. Ability to execute within instructor's test harness	5 pts.
Total:	20 pts.

Note:

When submitting code for review or help with problems, you must submit the pseudocode as well, and it must be kept up to date to reflect your current implementation. You must also keep your .h file documented. I will first look at those to identify problems, then any code feedback will be directed to how your code implements your design, not if your code is solving the problem.