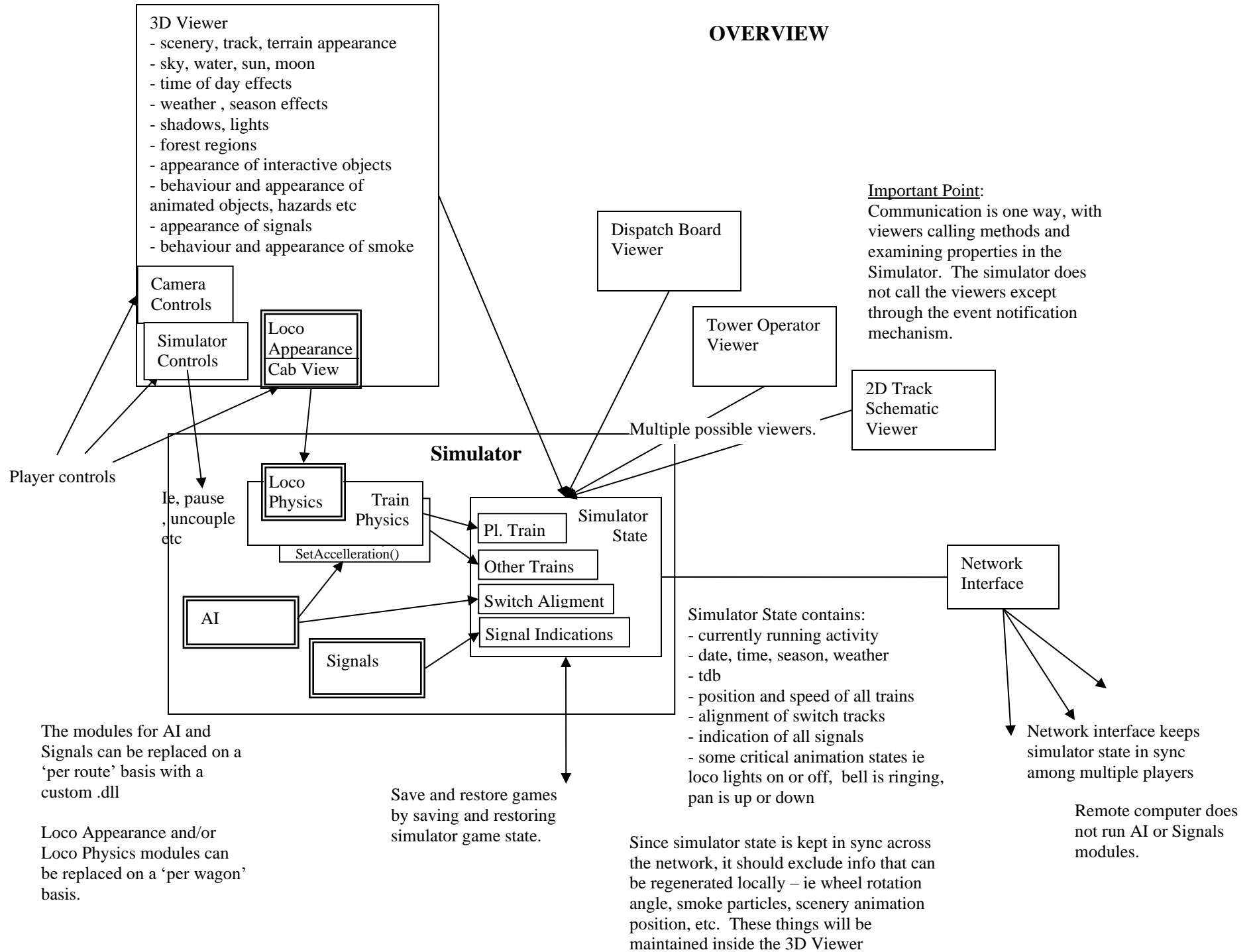
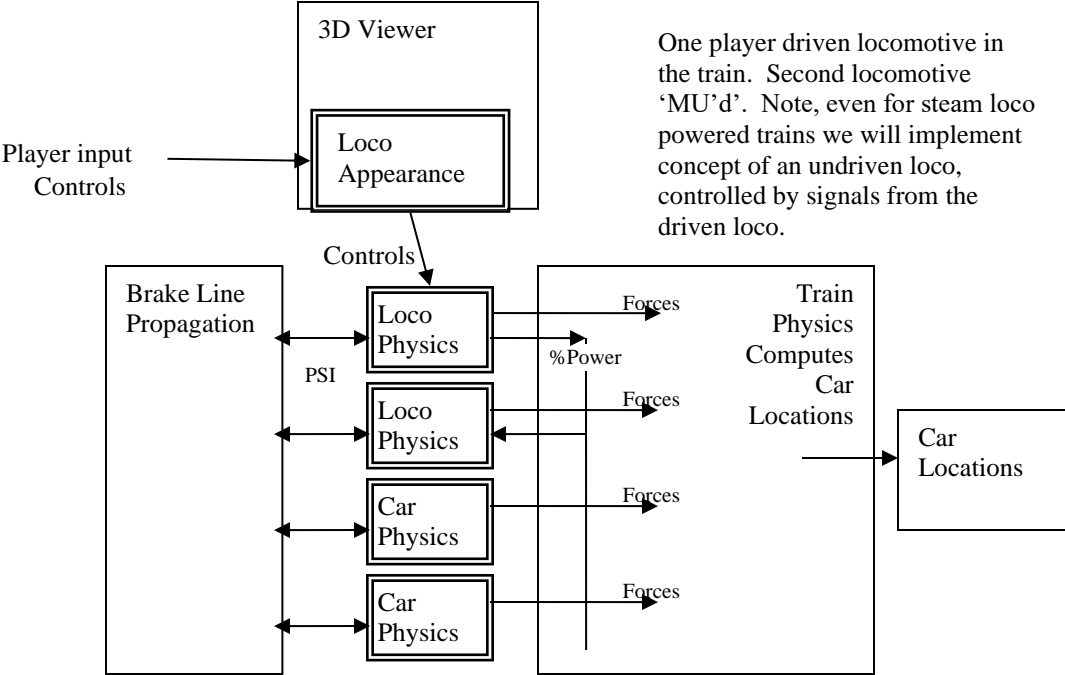


OVERVIEW

Important Point:
Communication is one way, with viewers calling methods and examining properties in the Simulator. The simulator does not call the viewers except through the event notification mechanism.



PLAYER CONTROLLED TRAIN

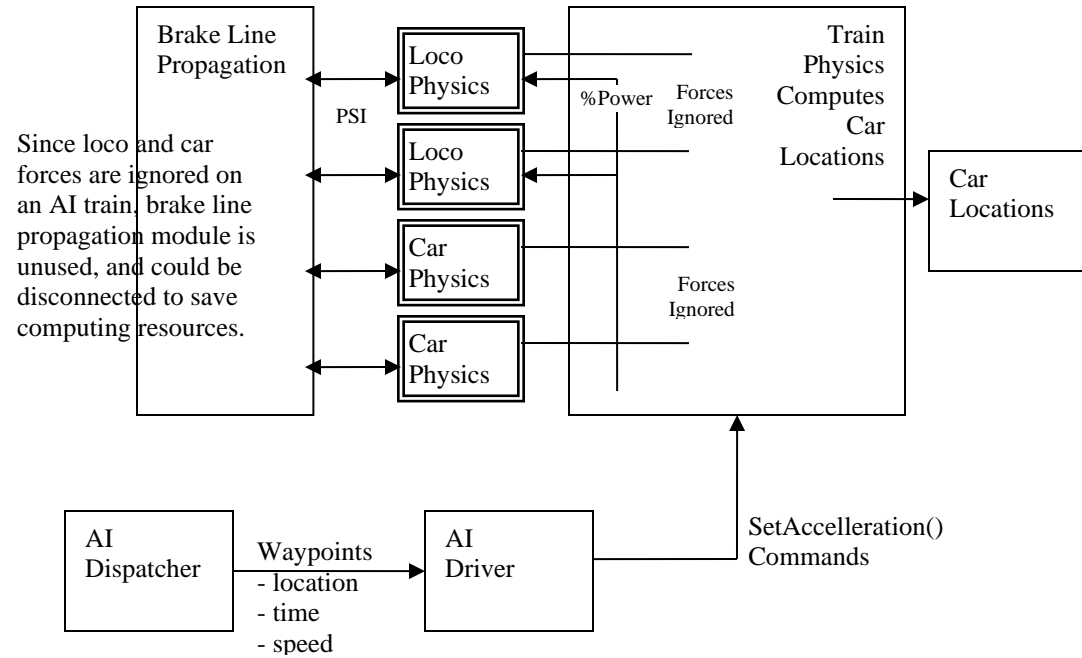


One player driven locomotive in the train. Second locomotive 'MU'd'. Note, even for steam loco powered trains we will implement concept of an undriven loco, controlled by signals from the driven loco.

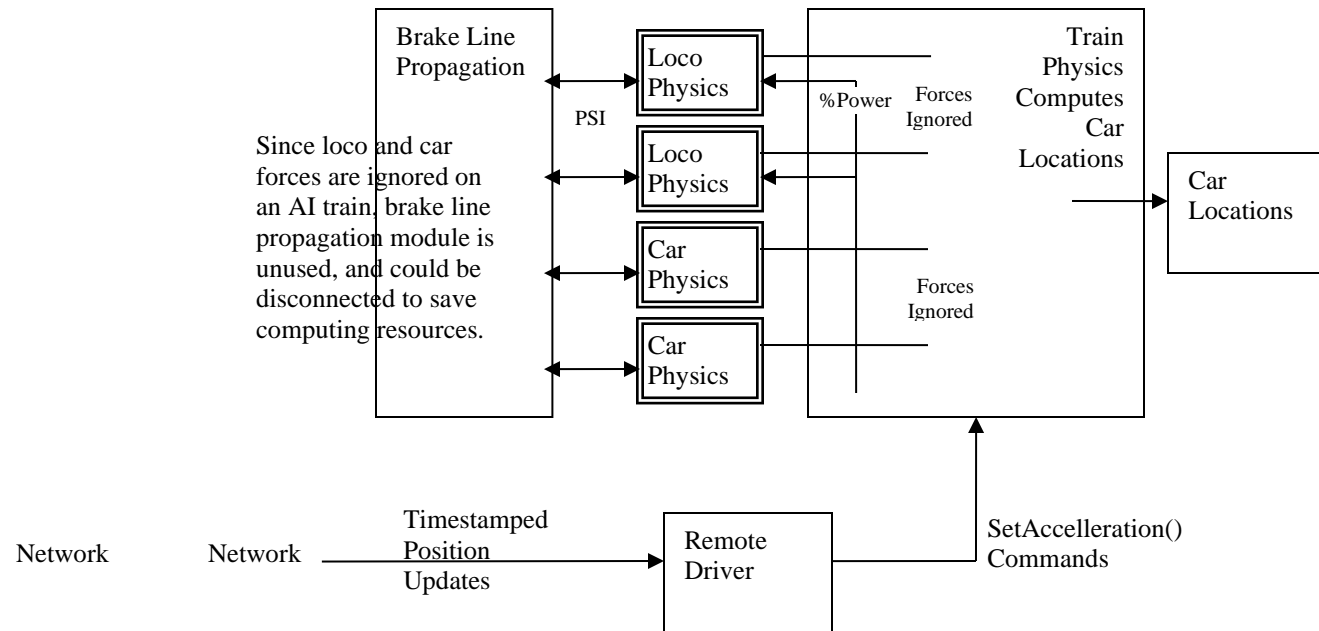
A Loco Physics module must be able to respond to control commands from its matching 3D viewer module., OR, %power commands. The %power commands is the mechanism used to control undriven loco's in a multiple unit train, to control a loco on an AI driven train, and to control a loco that is driven remotely on another PC. Note that in the latter two cases, since the force calculations are ignored, the purpose of notifying the loco physics module of the loco's power setting is only to ensure its sounds, smoke generation etc are appropriate for the power levels it is running at.

AI CONTROLLED TRAIN

AI Controlled Train. The train physics module sends %power signals to physics module to control animation etc of the loco under power. However, forces are ignored. To ensure deterministic behaviour, train motion follows acceleration commands from AI

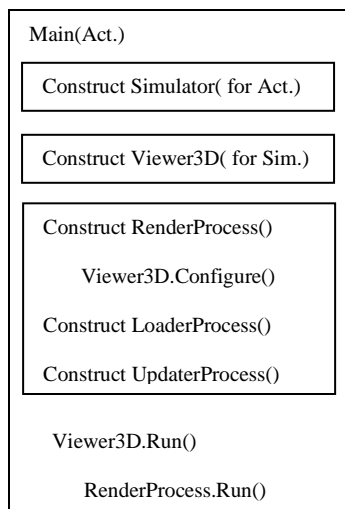


REMOTE CONTROLLED TRAIN



In a multiple player environment, each train is controlled by only one controlling computer. All other computers update the position of the train via the remote driver. The remote driver receives broadcasted timestamped position updates from the controlling computer. It compares these with where the train actually is on his computer, and issues setAcceleration commands as need .

STARTING PROGRAM



RenderProcess
(XNA Game Class)

STARTING 3D GRAPHICS SYSTEM

