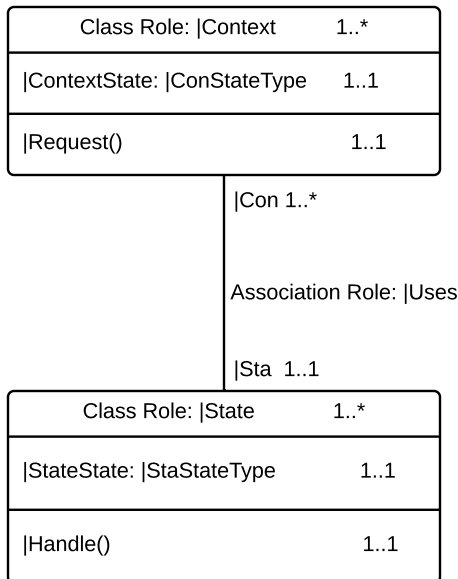
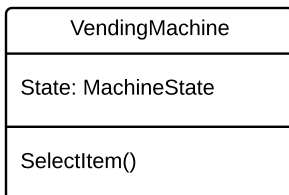


## Question 2

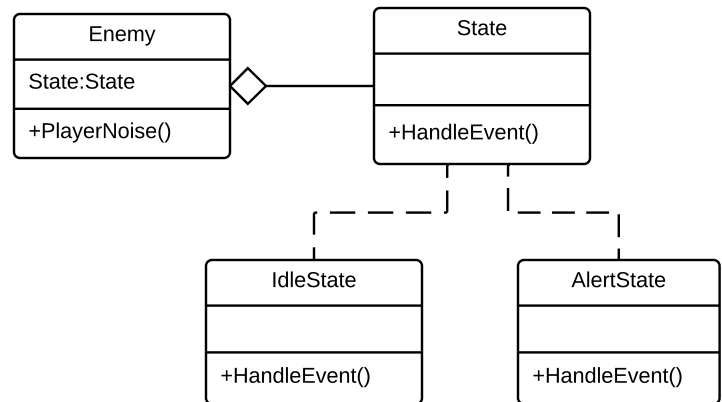
### State Pattern RBML



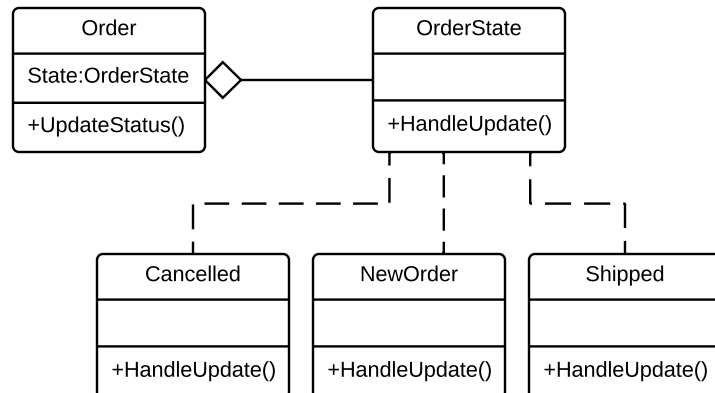
### ii. Near Instance (There are no states to use)



### i. Realization Example 1



### i. Realization Example 2



### iii Restrictive vs. Lenient Question

It is definately a fine line between restrictive and lenient, restrictive is very important if you are sending your meta model to a different company for instance, you would want it easy for them to understand and follow. However this also leave little to no room for change, so a meta model needs to be somewhat lenient. So I think the obvious solution would be to allow the inclusion of metamodel extensions, these could be possible changes that could be submitted for approval then possibly added to the existing model that way you could control the amount of changes leaving room for some amount of leniency. This idea of restrictive vs leniency is very important because without a certain amount of leniency it leaves programmers very little room to improve upon a design if the situation calls for it especially because no matter how much a design may be looked over, sometimes its not until a fresh set of eyes looks at something that issues may be discovered.