```
Model Commanding
 Parameters:
   subsytemType : {ApogeeBACnet, ApogeeBACnetFLN, 3rdPartyBACnet, Modbus, DesigoClassic, SystemOne, SystemOneAX100}
   pointType:  {BO, MO, AO, BV, MV, AV, BI, AI, HoldingReg1, HoldingReg2}
   commandType: {slider, buttons_plusMinus, textBox, buttonsOnOff, buttons3stages, NA}
   commandToPointValue : { readOnly, cancel, anyValue, higherByOne, lowerByOne, lowBoundary, highBoundary, lowBoundaryBeyond, highBoundaryBeyond}
```

Point type:
BO (Binary Output) or
BV (Binary Value)


```
Constraints:
// if pointType is BO or BV, you can only command it +1, -1, cancel, command type is fixed to OnOff and it's not readOnly
# (pointType = BO) || (pointType = BV) => ((commandToPointValue = higherByOne) || (commandToPointValue = lowerByOne) || (com
```

Point type:
MO (Multi-state output) or
MV (Multi-state value)


```
// if pointType is MO or MV, you can only command it +1 (and another), -1 (and another), cancel, command type is fixed
# (pointType = MO) || (pointType = MV) => ((commandToPointValue = higherByOne) || (commandToPointValue = lowerByOne) ||
```

Point type:
AO (Analog output) or
AV (Analog value)


```
// if pointType is AO or AV, commandType cannot be those used for BO and MO, and it's not readOnly
# (pointType = AO) || (pointType = AV) => (commandType != buttonsOnOff) && (commandType != buttons3stages) && (commandToP
// if commandType buttons + - (and AO or AV), commandToPointValue one higher or one lower
# ((pointType = AO) || (pointType = AV)) && (commandType = buttons_plusMinus) => (commandToPointValue = higherByOne) || (
// if commandType is slider, you cannot go beyond the boundaries and no need to test +1 -1
# ((pointType = AO) || (pointType = AV)) && (commandType = slider) => (commandToPointValue != lowBoundaryBeyond) && (comm
```