

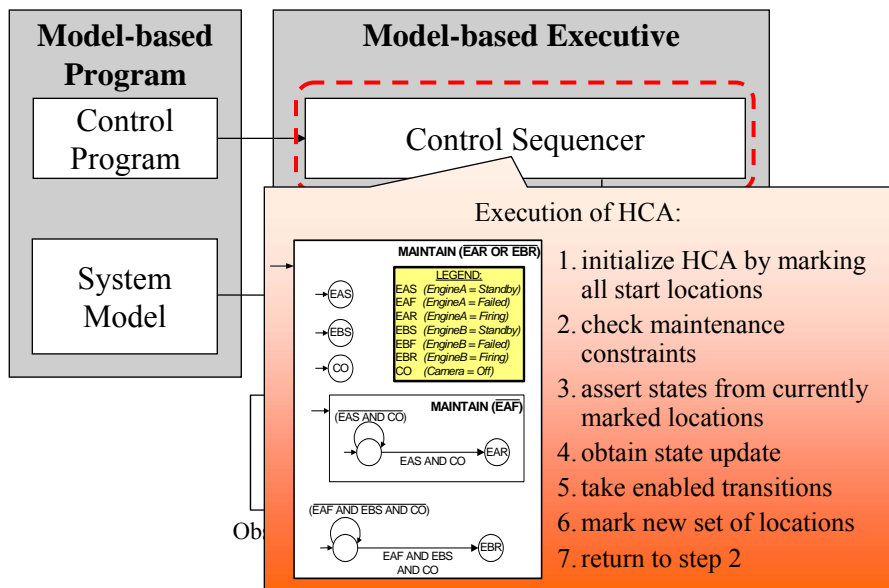
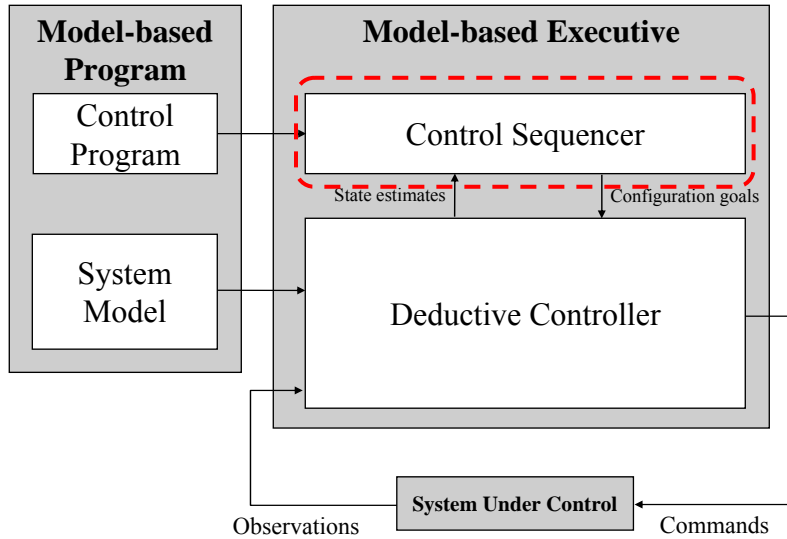
# Model-based Execution

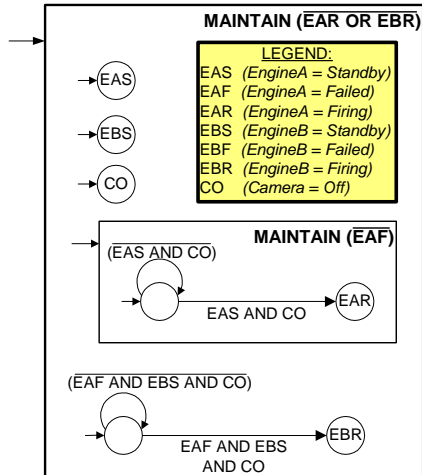
1

## Outline

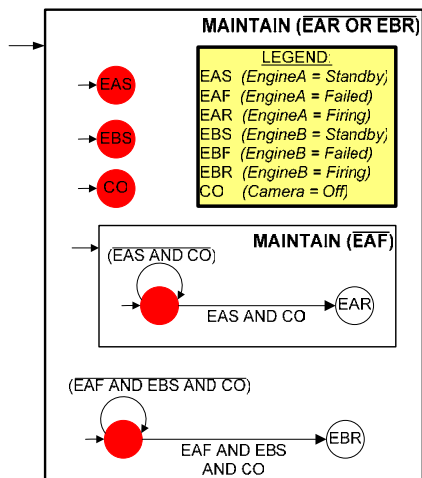
- Introduction & Overview
- Model-based Programming
- Execution of Model-based Programs
  - Scenario revisited
  - Control Sequencer
  - Deductive Controller (Mode Estimation & Mode Reconfiguration)
  - Model-based Execution Semantics
- Fundamentals of Model-based Reasoning
- Modeling via State Analysis
- Advanced Methods
- Conclusion

2

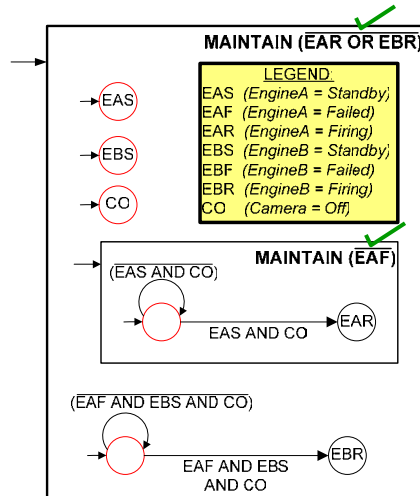




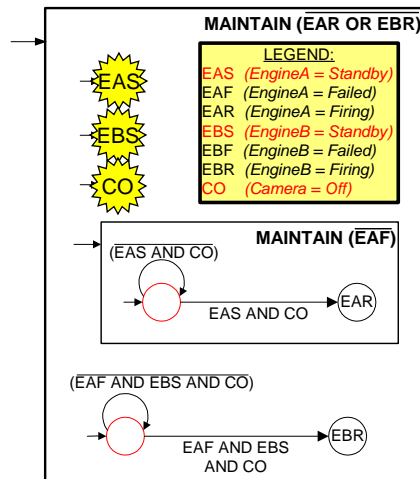
Nominal (i.e. fault-free) orbital insertion scenario



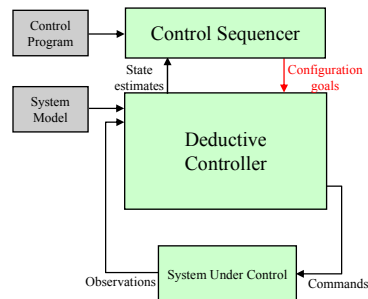
- initialize HCA by marking all start locations

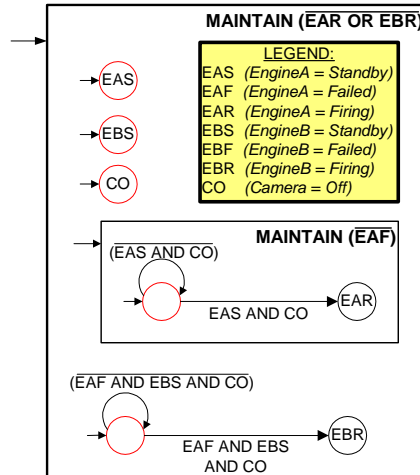


- initialize HCA by marking all start locations
- check maintenance constraints

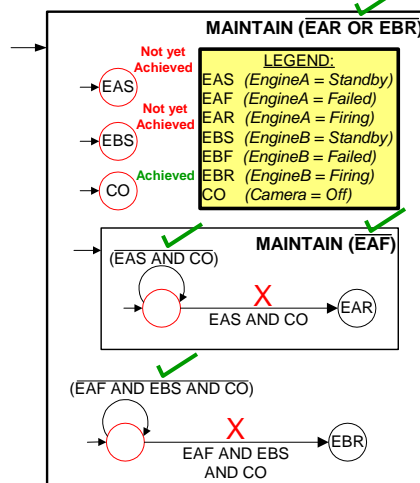
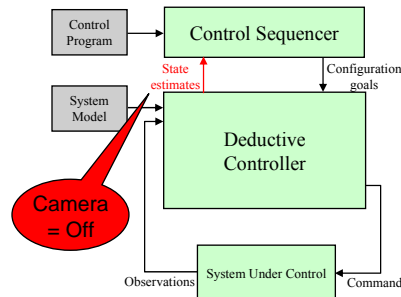


- initialize HCA by marking all start locations
- check maintenance constraints
- assert states from currently marked locations

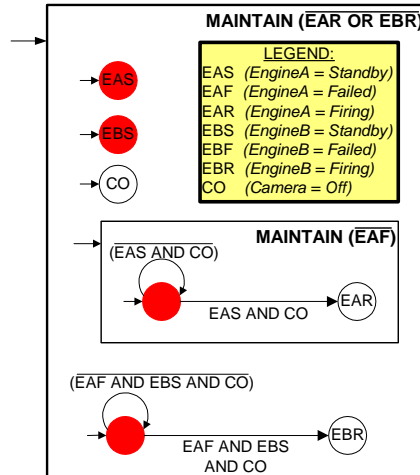




- initialize HCA by marking all start locations
- check maintenance constraints
- assert states from currently marked locations
- obtain state update

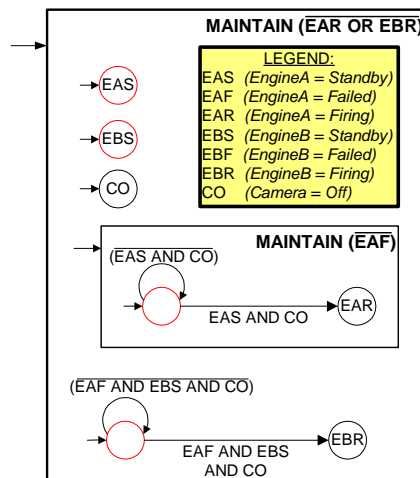


- initialize HCA by marking all start locations
- check maintenance constraints
- assert states from currently marked locations
- obtain state update
- take enabled transitions:
  - location's state assignment achieved
  - transition and maintenance conditions currently hold true



- initialize HCA by marking all start locations
- check maintenance constraints
- assert states from currently marked locations
- obtain state update
- take enabled transitions:
  - location's state assignment achieved
  - transition and maintenance conditions currently hold true
- mark new set of locations

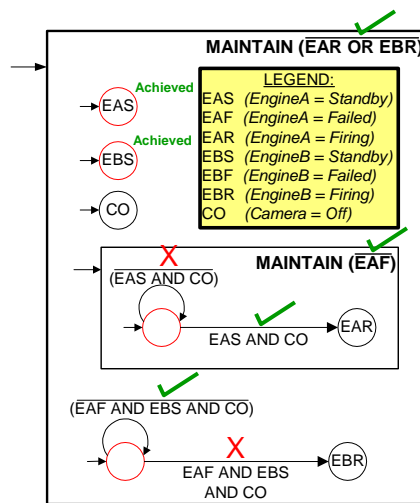
11



- (*EngineA = Standby*) & (*EngineB = Standby*) achieved in this step

12

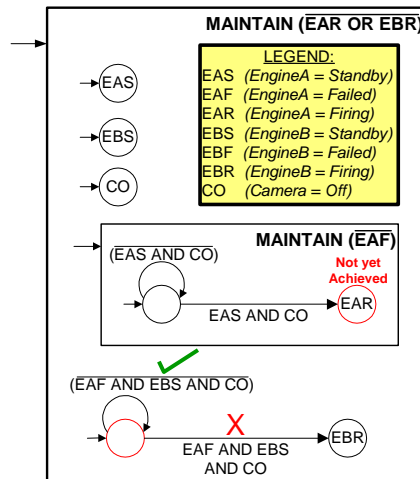
## Executing HCA - Step 2



- (EngineA = Standby) & (EngineB = Standby) achieved in this step
- two execution threads terminated & two transitions enabled

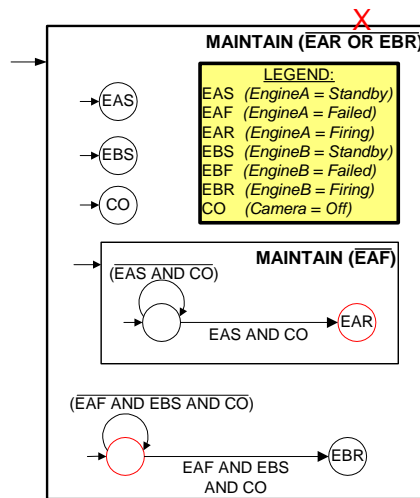
13

## Executing HCA - Step 3

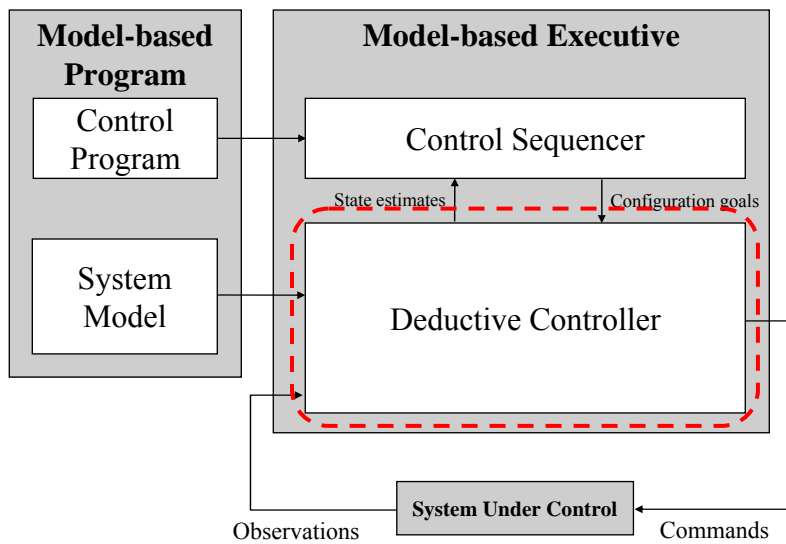


- (EngineA = Firing) asserted in this step, but not yet achieved

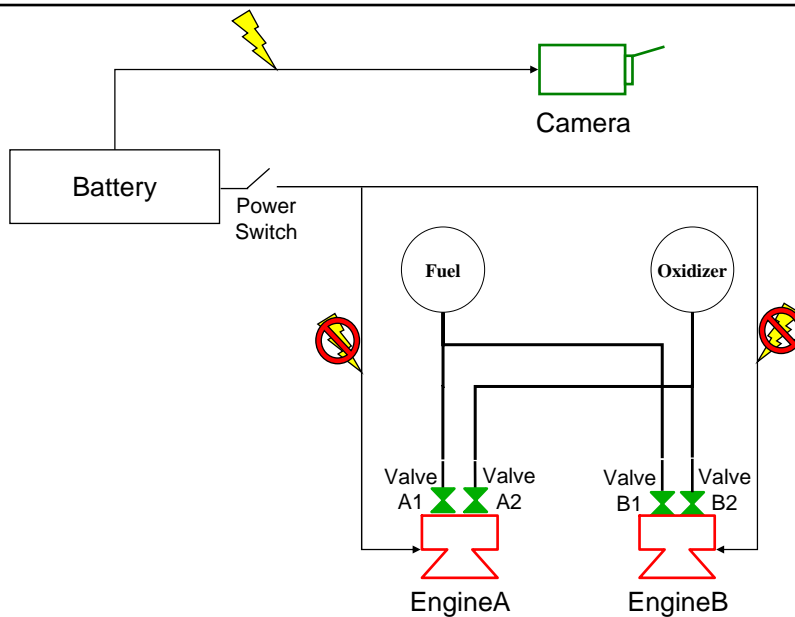
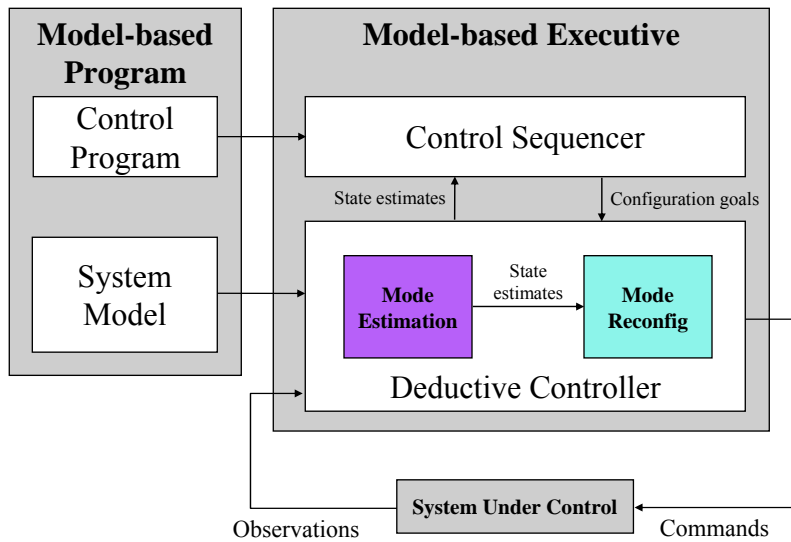
14

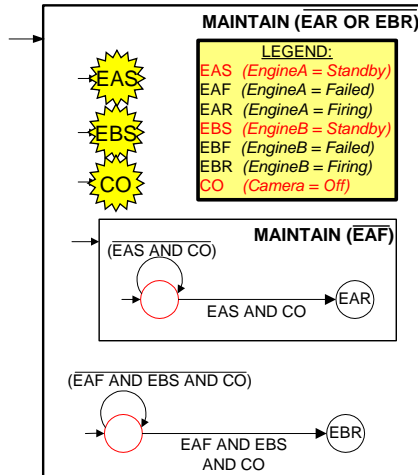


- (EngineA = Firing) achieved in this step
- maintenance condition violated, HCA block exited

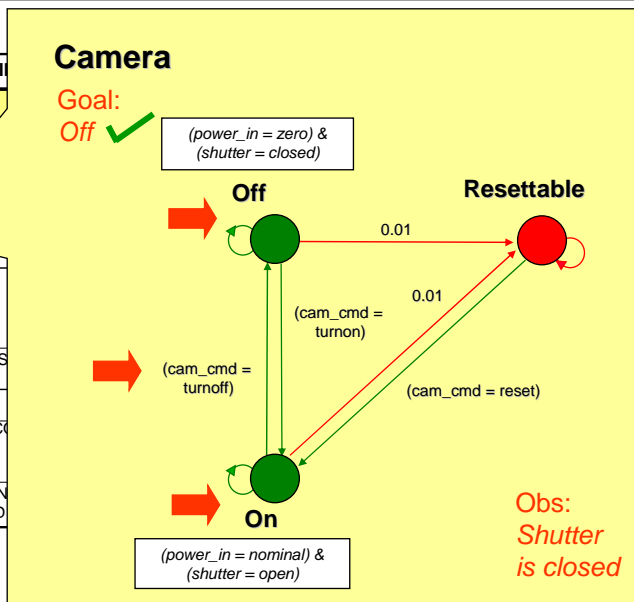
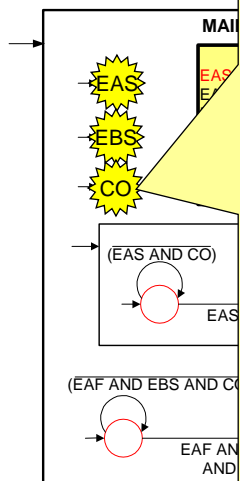
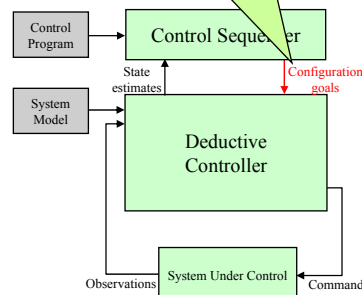


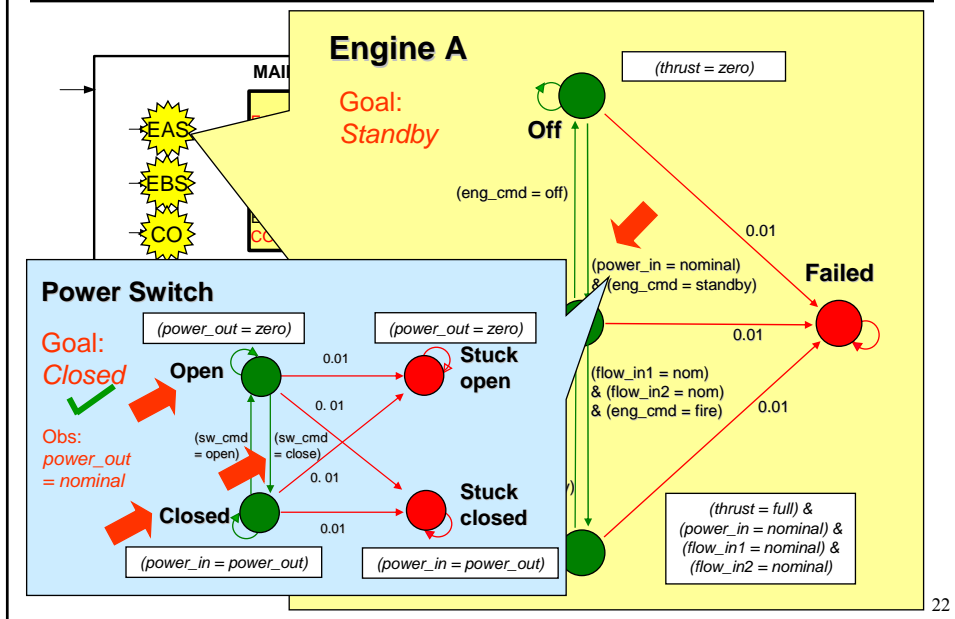
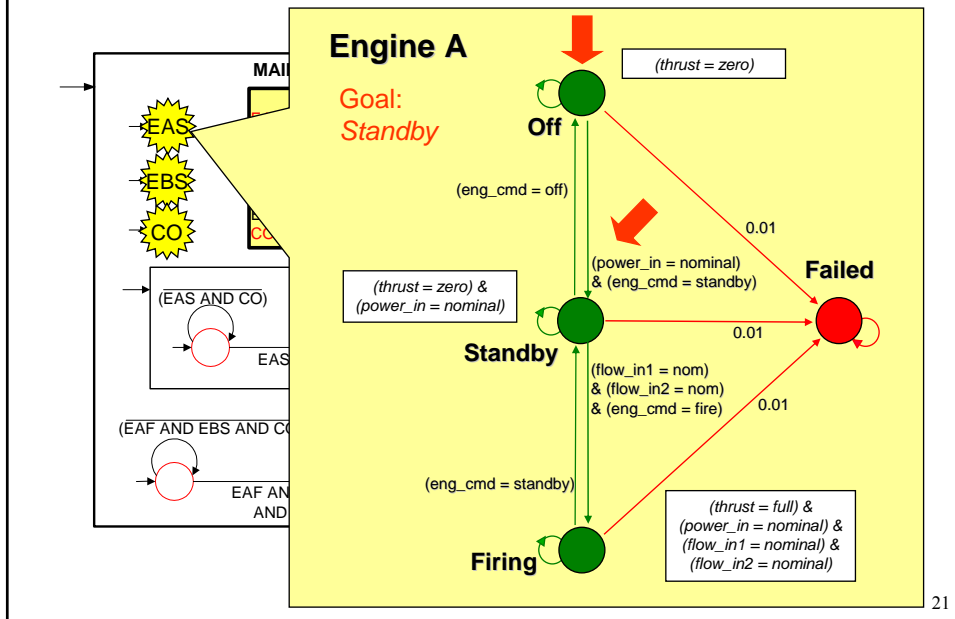


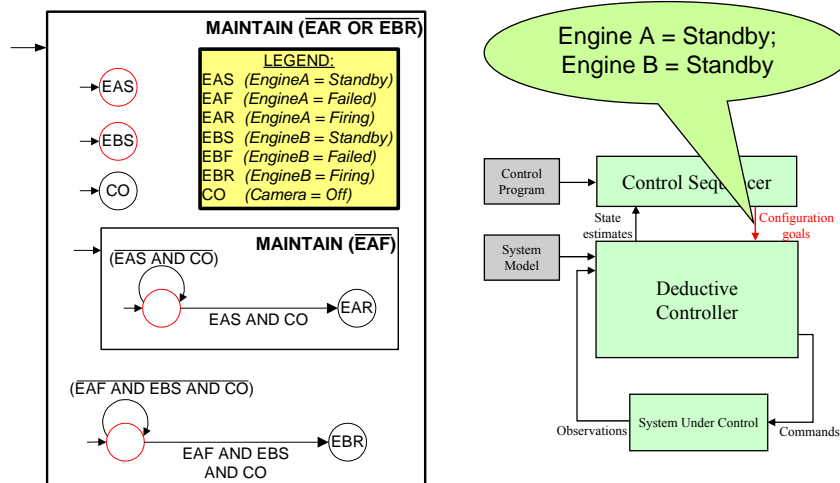
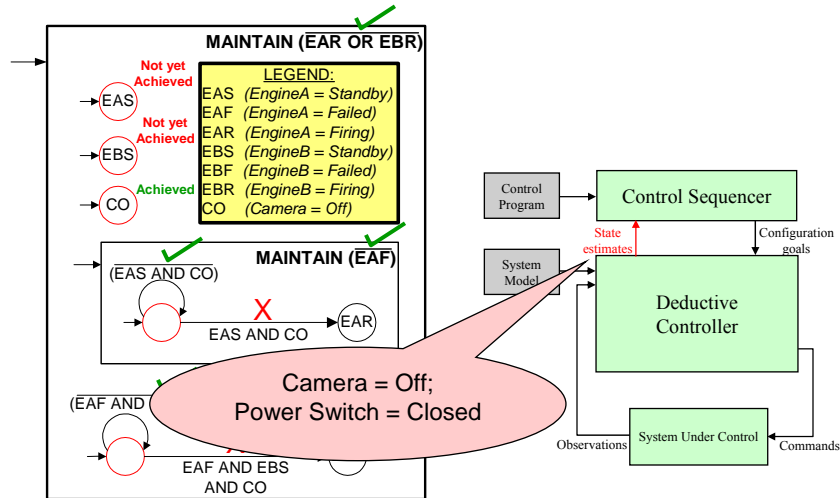


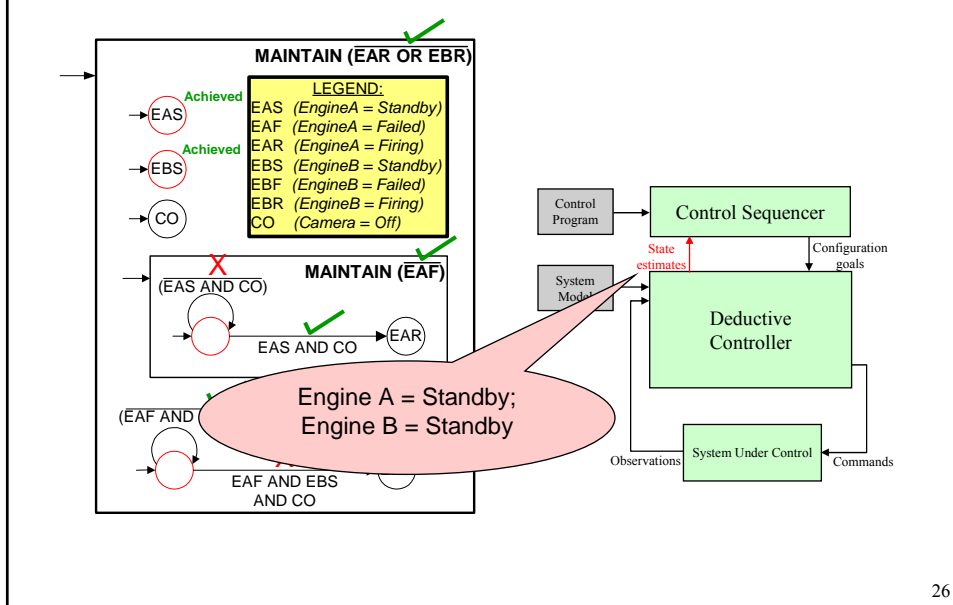
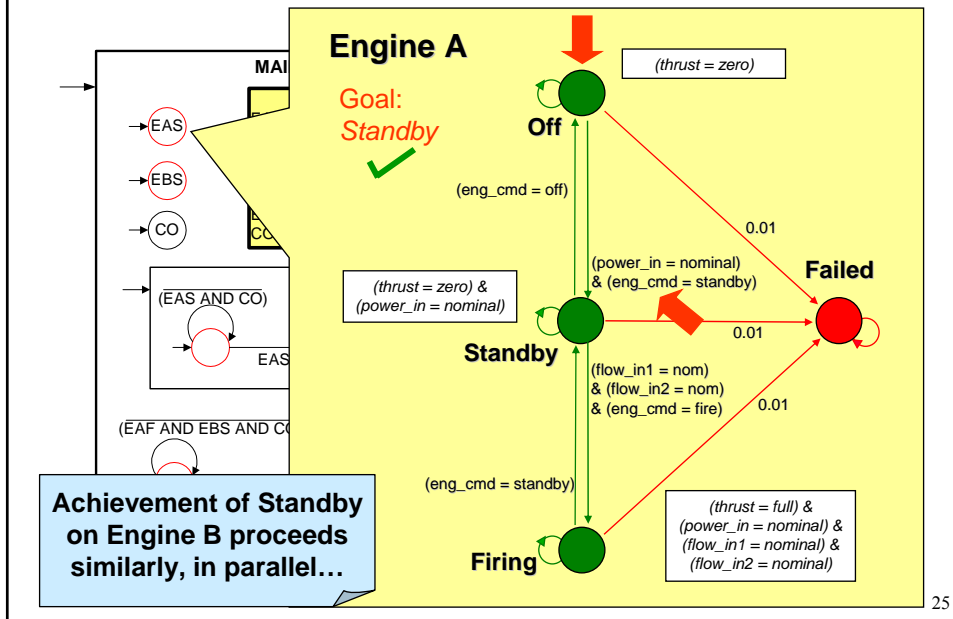


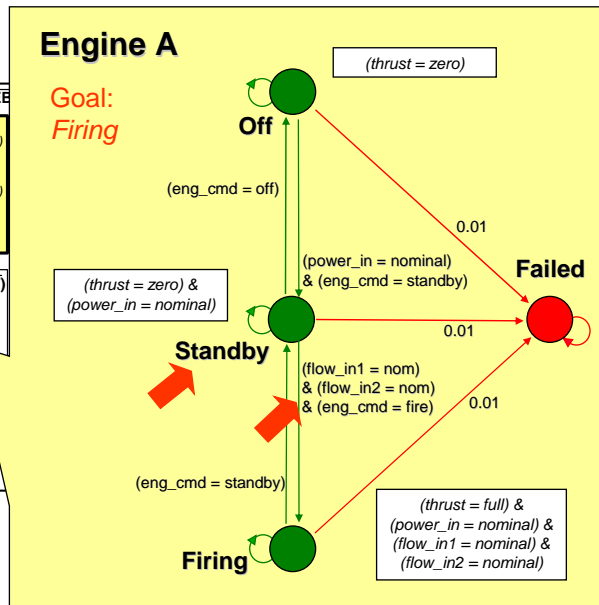
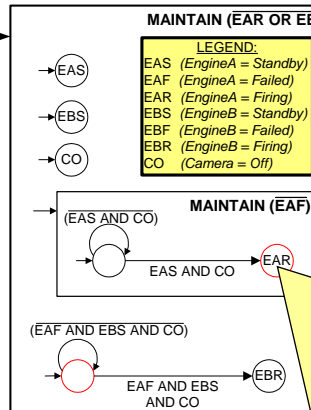
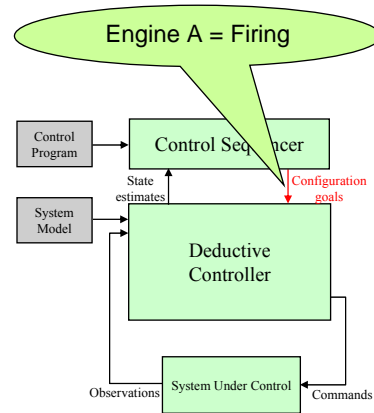
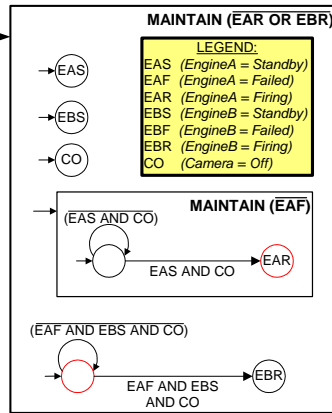
Camera = Off;  
Engine A = Standby;  
Engine B = Standby

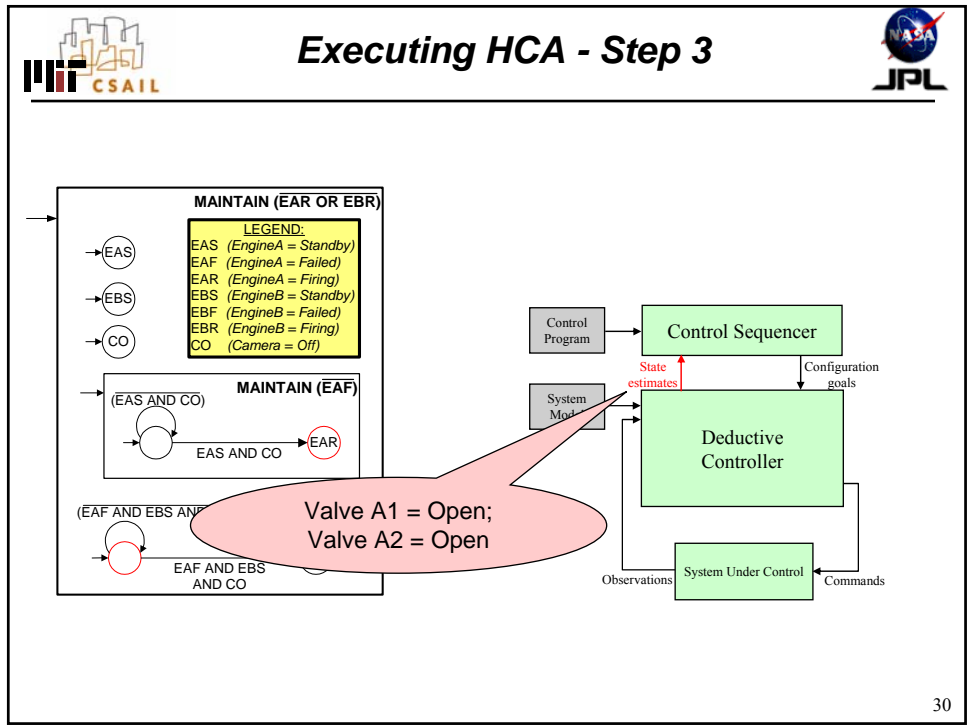
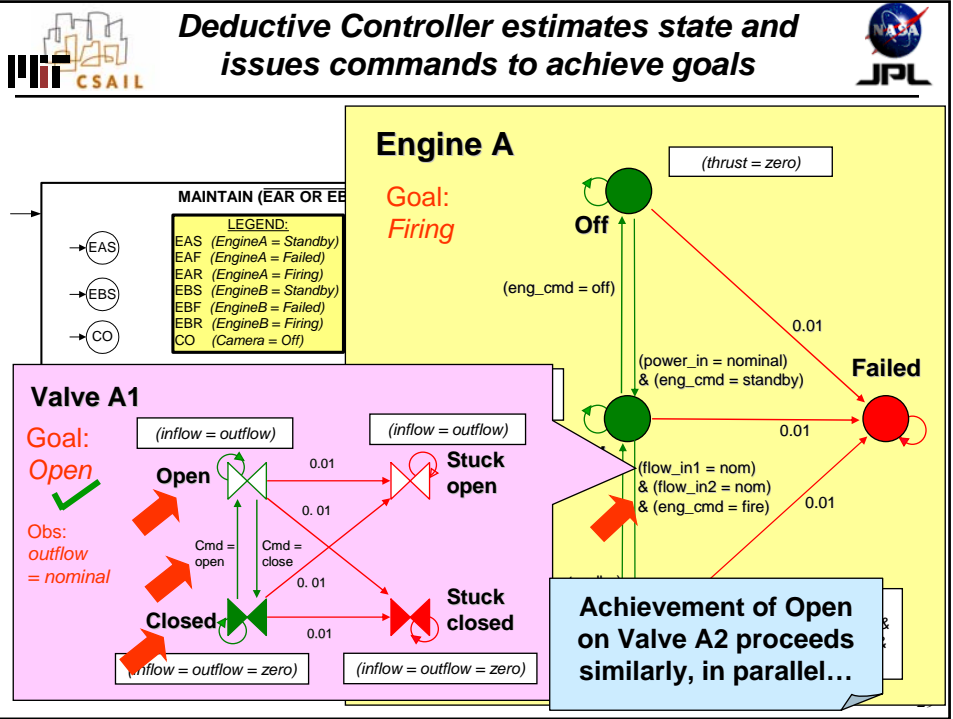


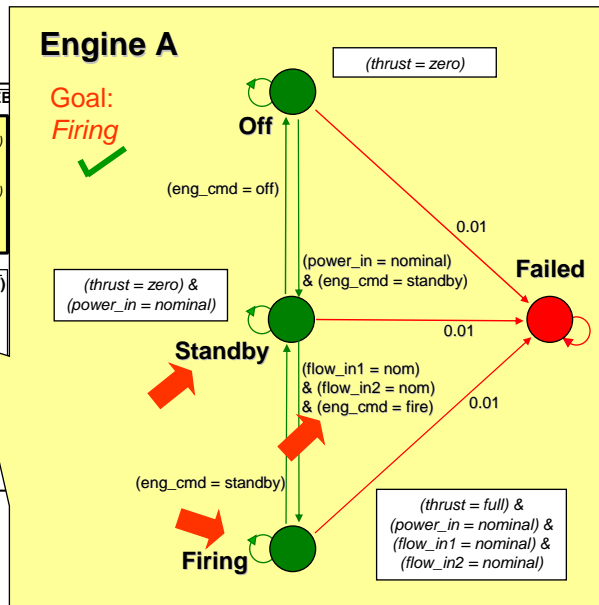
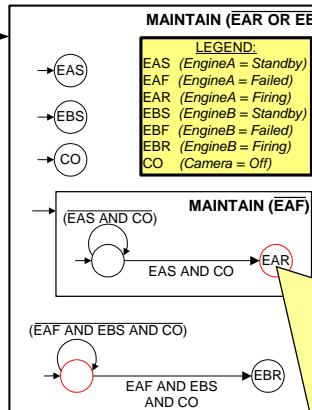
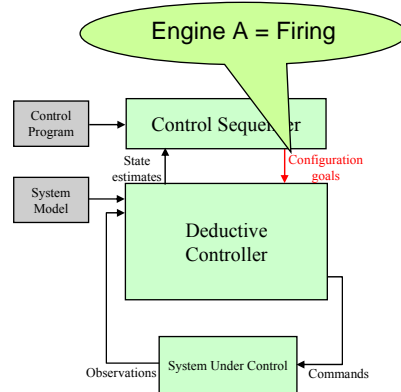
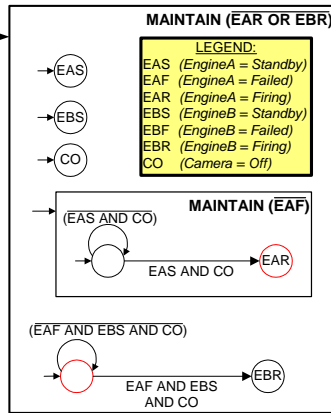




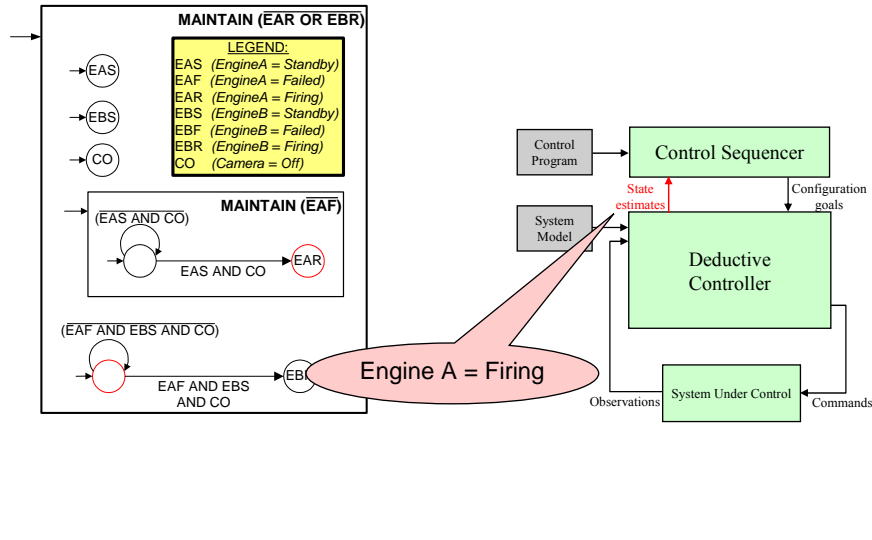




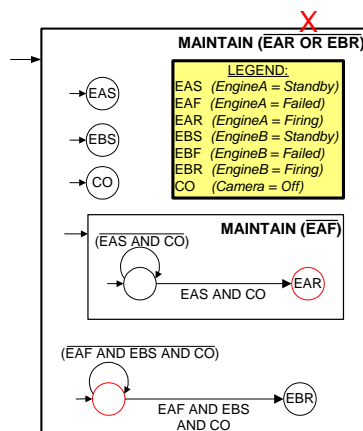






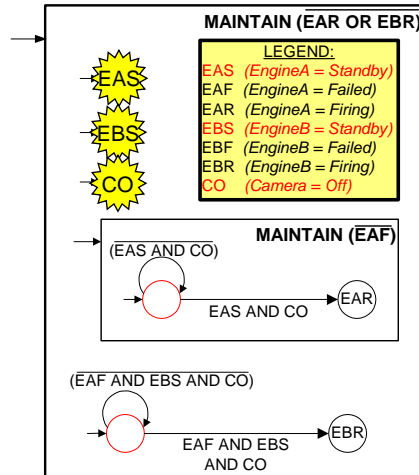


33

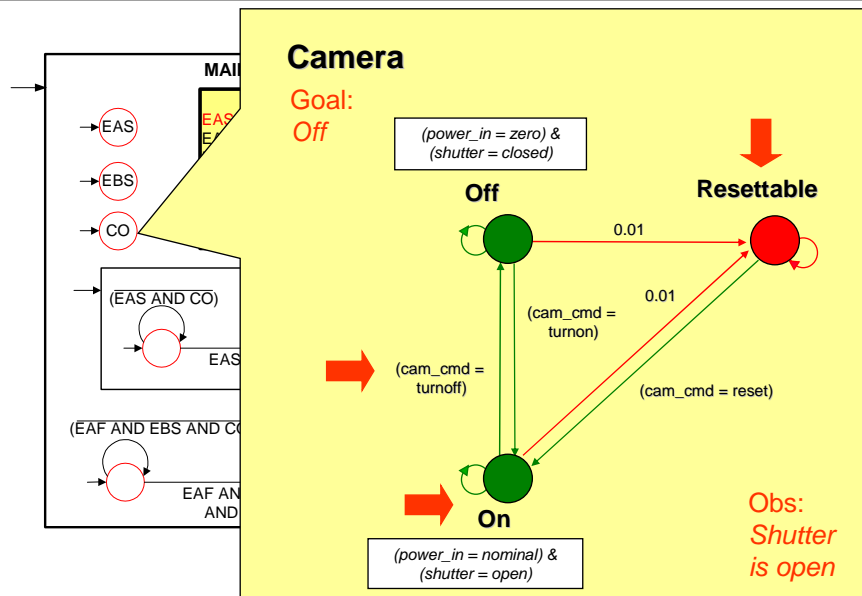
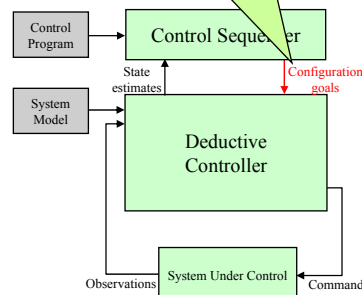


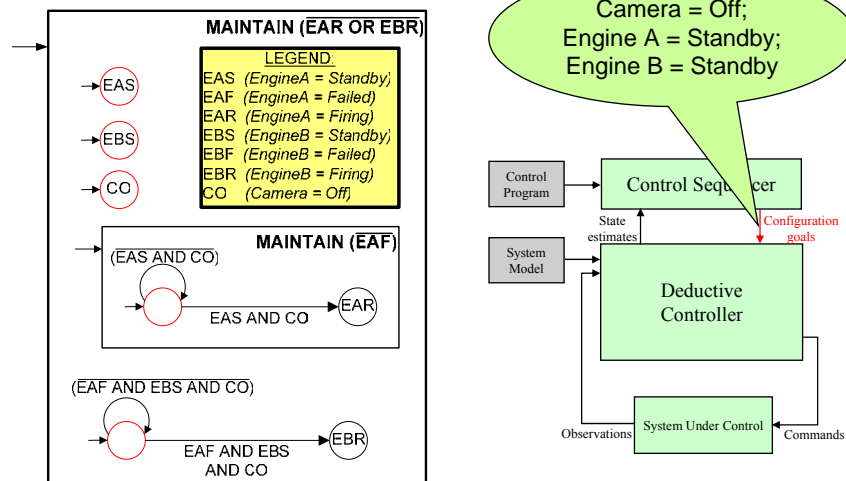
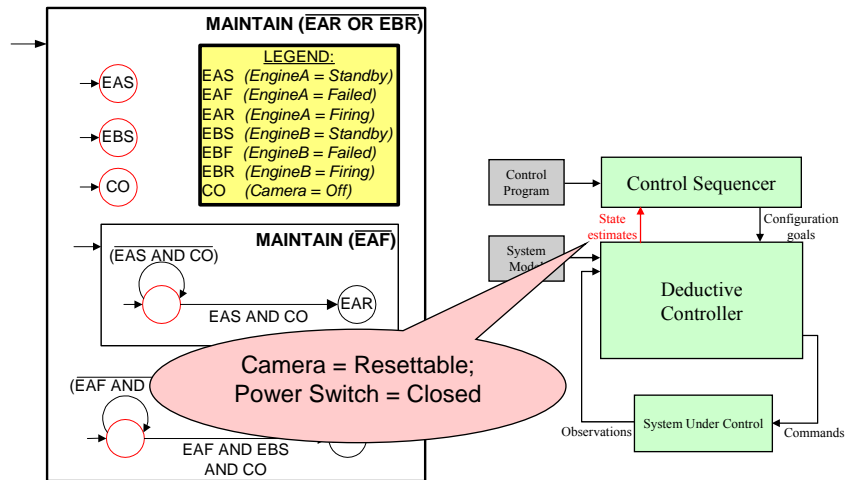
- (EngineA = Firing) achieved in this step
- maintenance condition violated, HCA block exited

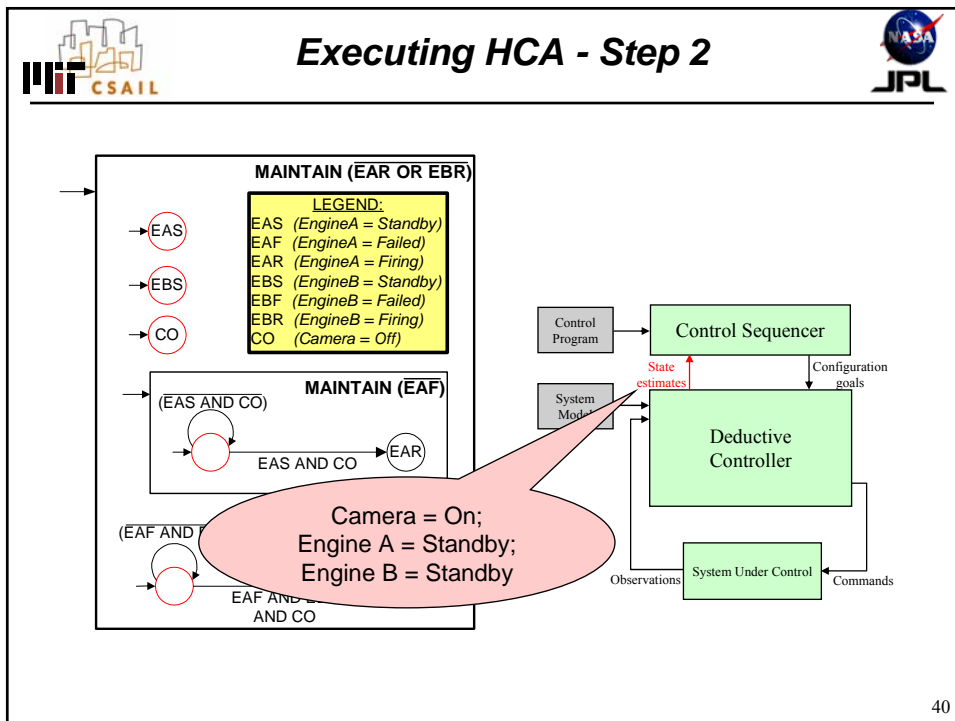
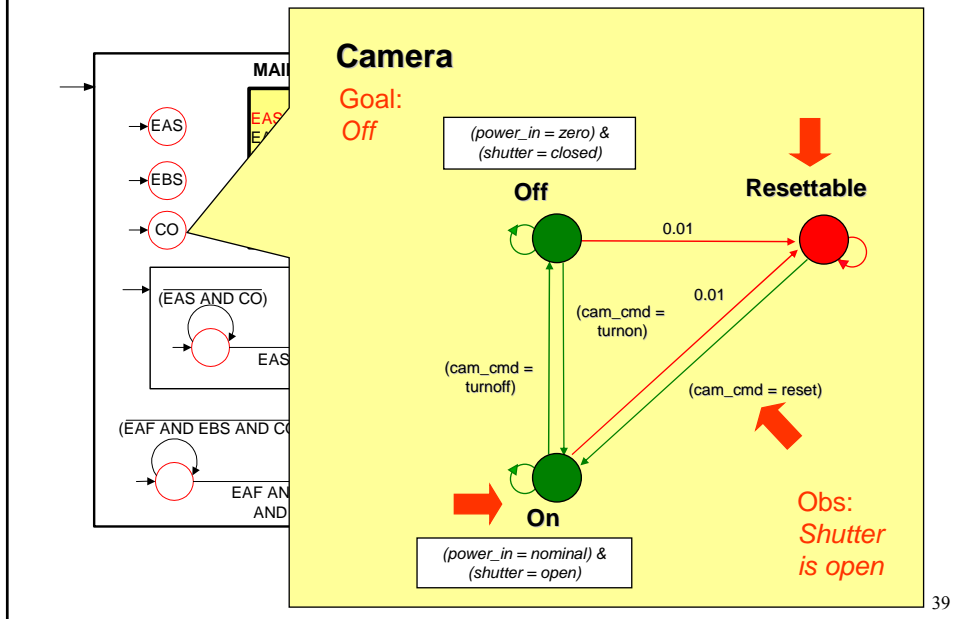
34

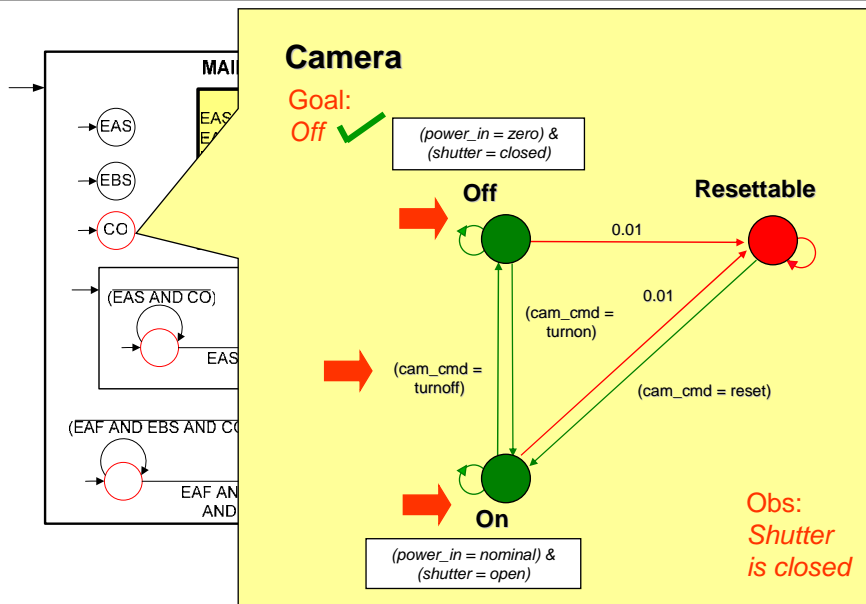
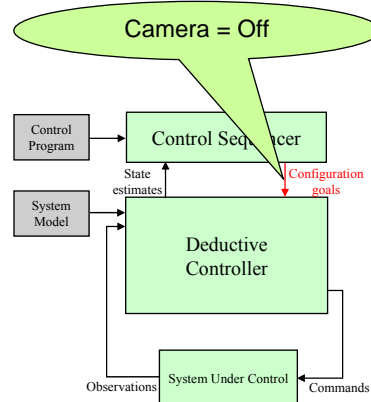
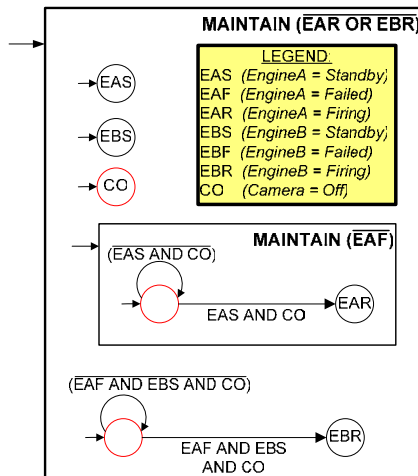


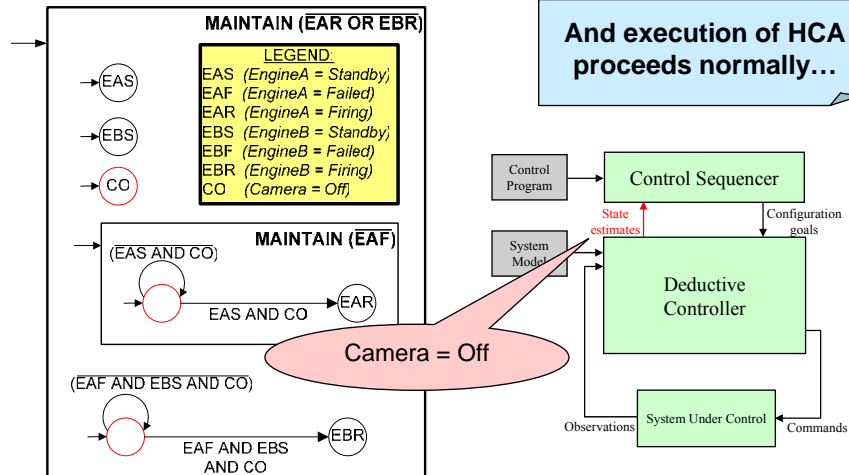
Camera = Off;  
Engine A = Standby;  
Engine B = Standby











43

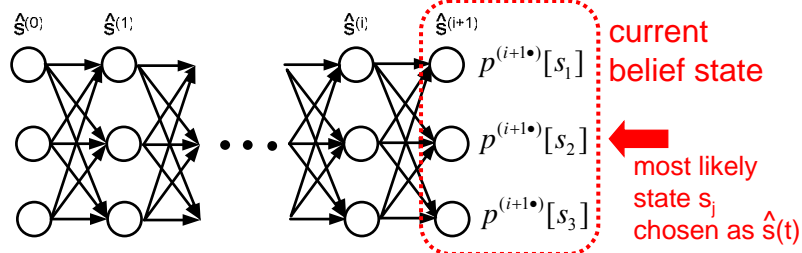
- Now that we've informally described Model-based Execution and illustrated how it works...
- We can now proceed with a (brief) formal semantic specification of Model-based Execution
- This follows from the semantic representations of the Model-based Program that we presented earlier:

$$CP = \langle L_{cp}, \lambda_{cp}, \tau_{cp}, g_{cp}, \Sigma_s \rangle$$

$$SM = \langle \Sigma, T, P_{\Theta}, P_T, P_O, R \rangle$$

44

- Given latest commands and observations, what is the most likely current state?
- Belief state update to estimate state for POMDPs:

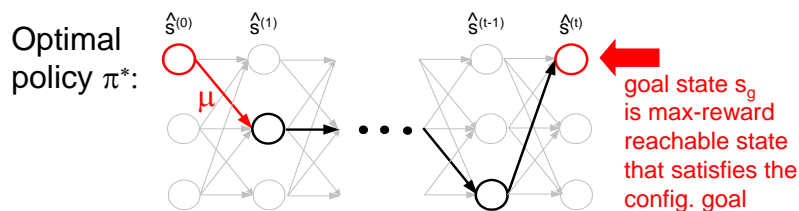


$$p^{*(i+1)}[s_l] = \sum_{k=1}^n p^{*(i)}[s_k] P_T(s_l | s_k, \mu)$$

$$p^{*(i+1)*}[s_l] = p^{*(i+1)}[s_l] \frac{P_O(o^{(i+1)} | s_l)}{\sum_{k=1}^n p^{*(i+1)}[s_k] P_O(o^{(i+1)} | s_k)}$$

45

- Given current belief state and configuration goal, what is the first control action from a policy that maximizes expected reward?



- Solve Bellman equation to compute optimal policy for POMDPs:

$$V^*(s) = \max_{\pi} E \left[ \sum_{i=1}^{\infty} \gamma^i r_i \right]$$

$$\pi^*(s) = \arg \max_{\mu} \left[ R_g(s) + \gamma \sum_{s' \in S} P_T(s' | s, \mu) V^*(s') \right]$$

46

- Interleaving model of execution  
*cycle = discrete event + continuous phase*
- Legal execution of MBP:

Cycle start time	$t_0$	$t_1$	...
Plant state	$\hat{s}_0$		
Pgm location	$l_0$		

Such that:

1. initial conditions are valid
2. next state is legal
3. next program location is legal

$$P_{\Theta}(\hat{s}_0) > 0$$

$$l_0 = \lambda_{cp}$$

47

- Interleaving model of execution  
*cycle = discrete event + continuous phase*
- Legal execution of MBP:

Cycle start time	$t_0$	$t_1$	...
Plant state	$\hat{s}_0 \rightarrow \hat{s}_1$		
Pgm location	$l_0$		

Such that:

1. initial conditions are valid
2. next state is legal
3. next program location is legal

$$g_0 = g_{cp}(l_0)$$

$$\mu_0 = MR(SM, \hat{s}_0, g_0)$$

$$\hat{s}_1 = ME(SM, \hat{s}_0, \mu_0, o_1)$$

48



- Interleaving model of execution  
*cycle = discrete event + continuous phase*
- Legal execution of MBP:

Cycle start time	$t_0$	$t_1$	...
Plant state	$\hat{s}_0 \rightarrow \hat{s}_1$		
Pgm location	$l_0 \rightarrow l_1$		

Such that:

1. initial conditions are valid
2. next state is legal
3. next program location is legal

$$l_1 = \tau_{cp}(l_0, \hat{s}_0)$$

49

- Interleaving model of execution  
*cycle = discrete event + continuous phase*
- Legal execution of MBP:

Cycle start time	$t_0$	$t_1$	...
Plant state	$\hat{s}_0 \rightarrow \hat{s}_1$	$\hat{s}_1 \rightarrow \hat{s}_2$	...
Pgm location	$l_0 \rightarrow l_1$	$l_1 \rightarrow l_2$	...

Such that:

1. initial conditions are valid
2. next state is legal
3. next program location is legal

50



## Implementation Approximations



### Mode Estimation:

- Full belief state update is computationally infeasible
- Assume probability of a few most-likely states dominates probability of other possible states
- Track a limited set of most-likely states, from one cycle to the next

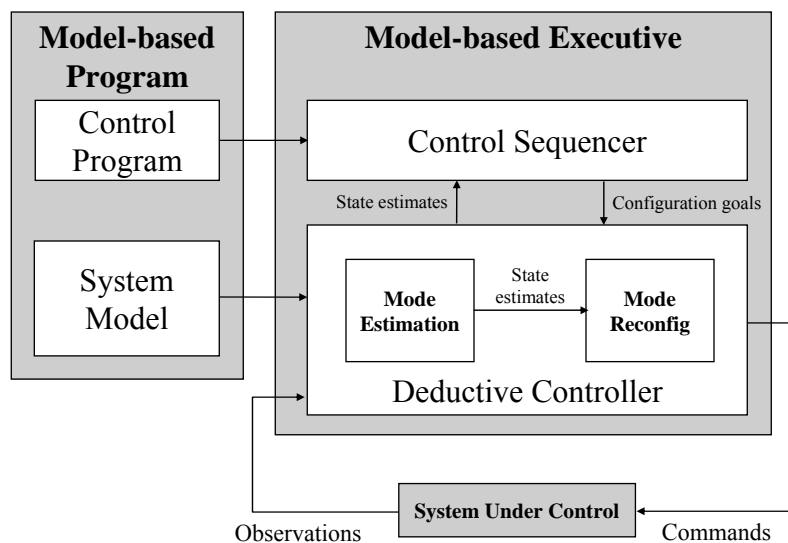
### Mode Reconfiguration:

- Assume probability of nominal behavior dominates off-nominal
- Assume reward of being in goal state dominates reward of getting to goal state
- Perform MR in 2 steps:
  - **Goal Interpretation:** find the *max-reward goal state*, reachable via *nominal transitions*, that satisfies the configuration goal
  - **Reactive Planning:** returns series of control actions that achieve the goal state

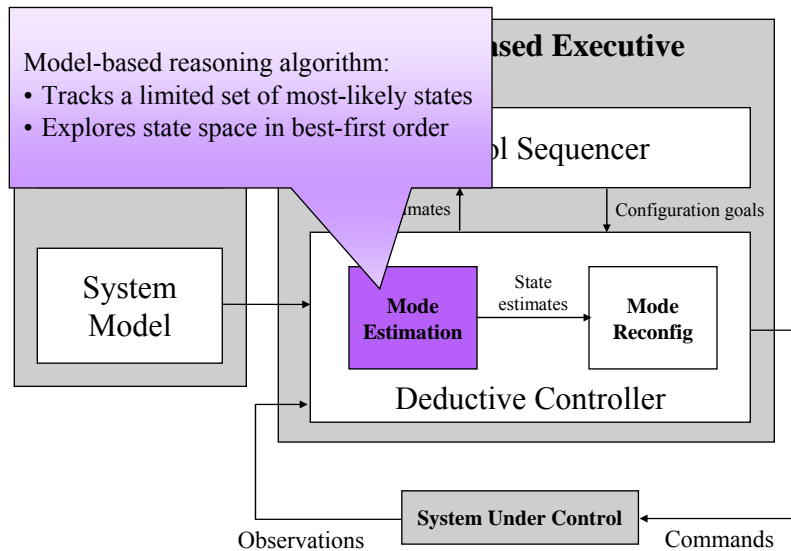
51



## Deductive Controller Implementation

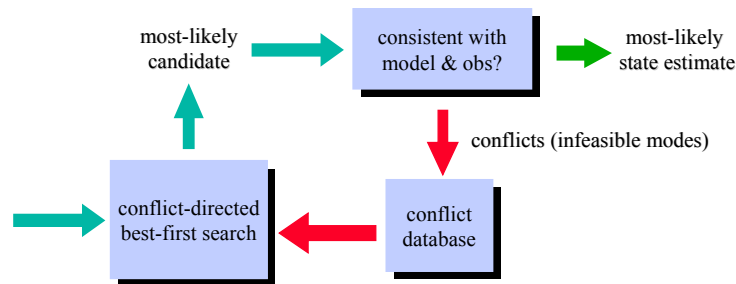


52



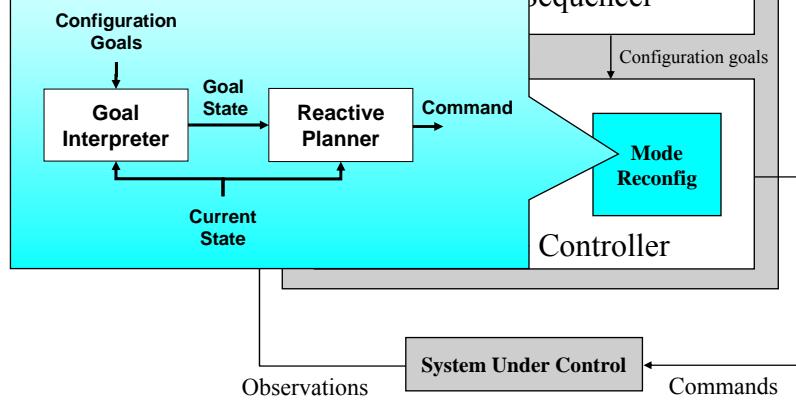
- Formulate Optimal Constraint Satisfaction Problem (OCSP), to identify “k-best” extensions to current trajectories (“shortest path” from set of current possible states to next possible states)
  - OCSP  $\langle \mathbf{x}, f, C \rangle$
  - *decision vars*  $\mathbf{x}$ , such that  $\text{dom}[\mathbf{x}_j] = \text{reachable target modes}$
  - *objective function*  $f(\mathbf{x}) = \text{prior probability of state } \mathbf{x}$ , i.e.:
 
$$\prod_j P_{T_j}(x_j | s^{(i)}, \mu^{(i)})$$
  - *constraint*  $C(\mathbf{x})$ , such that  $\mathbf{x} \wedge C_{Mx} \wedge O^{(i+1)}$  is consistent

- Formulate Optimal Constraint Satisfaction Problem (OCSP), to identify “k-best” extensions to current trajectories (“shortest path” from set of current possible states to next possible states)
- Solve using OPSAT engine

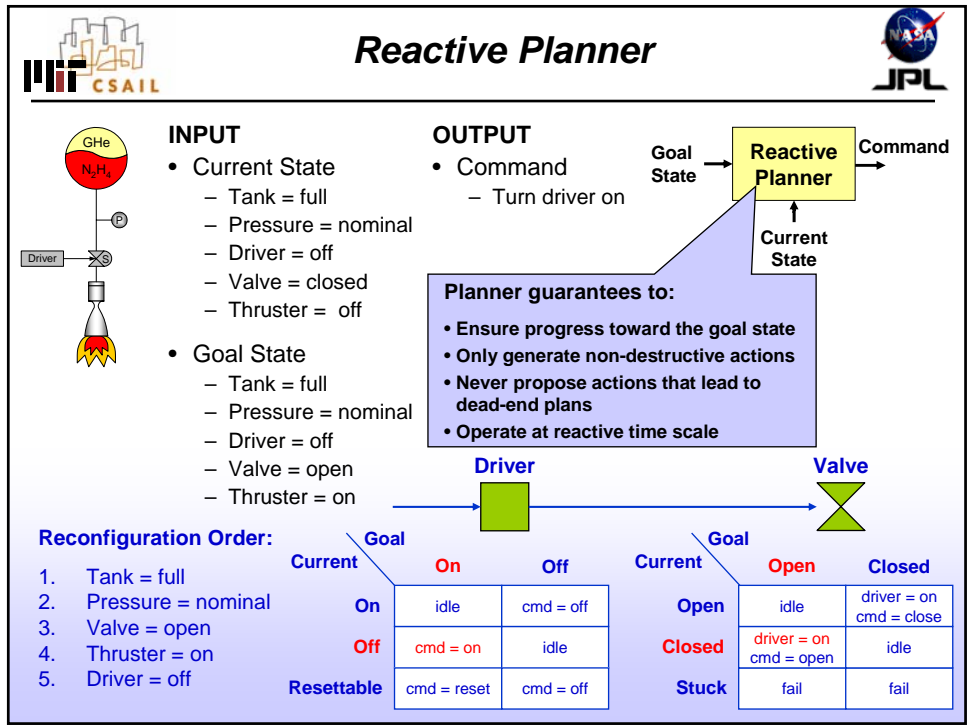
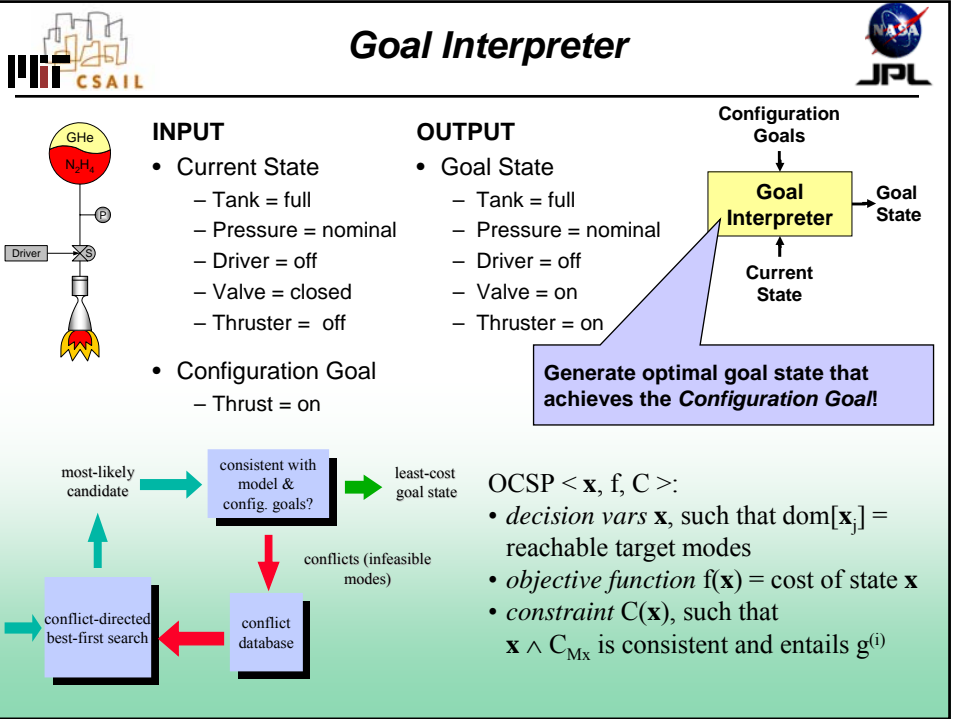


55

Model-based reasoning algorithms to compute a series of commands that progress the system towards a least-cost state that achieves the configuration goal, one command at a time.



56





## Summary



- Model-based Executive is made up of 2 components:
  - Control Sequencer
  - Deductive Controller
- The Deductive Controller performs two functions:
  - Mode Estimation
  - Mode Reconfiguration
- Model-based Execution has formal semantics:
  - legal state evolutions of a factored POMDP
  - intent expressed in the form of a deterministic automaton
- The implemented Model-based Executive overcomes the computational complexity of this problem by leveraging a few key assumptions and proven model-based reasoning algorithms

59



## Up Next...



- Fundamentals of Model-based Reasoning
  - A little historical context
  - A glimpse into the details behind the algorithms implemented in the Deductive Controller
  - A discussion of how we can improve the run-time performance of these algorithms by performing their most expensive reasoning steps off-line, at compile time

60