CMS – State Machine

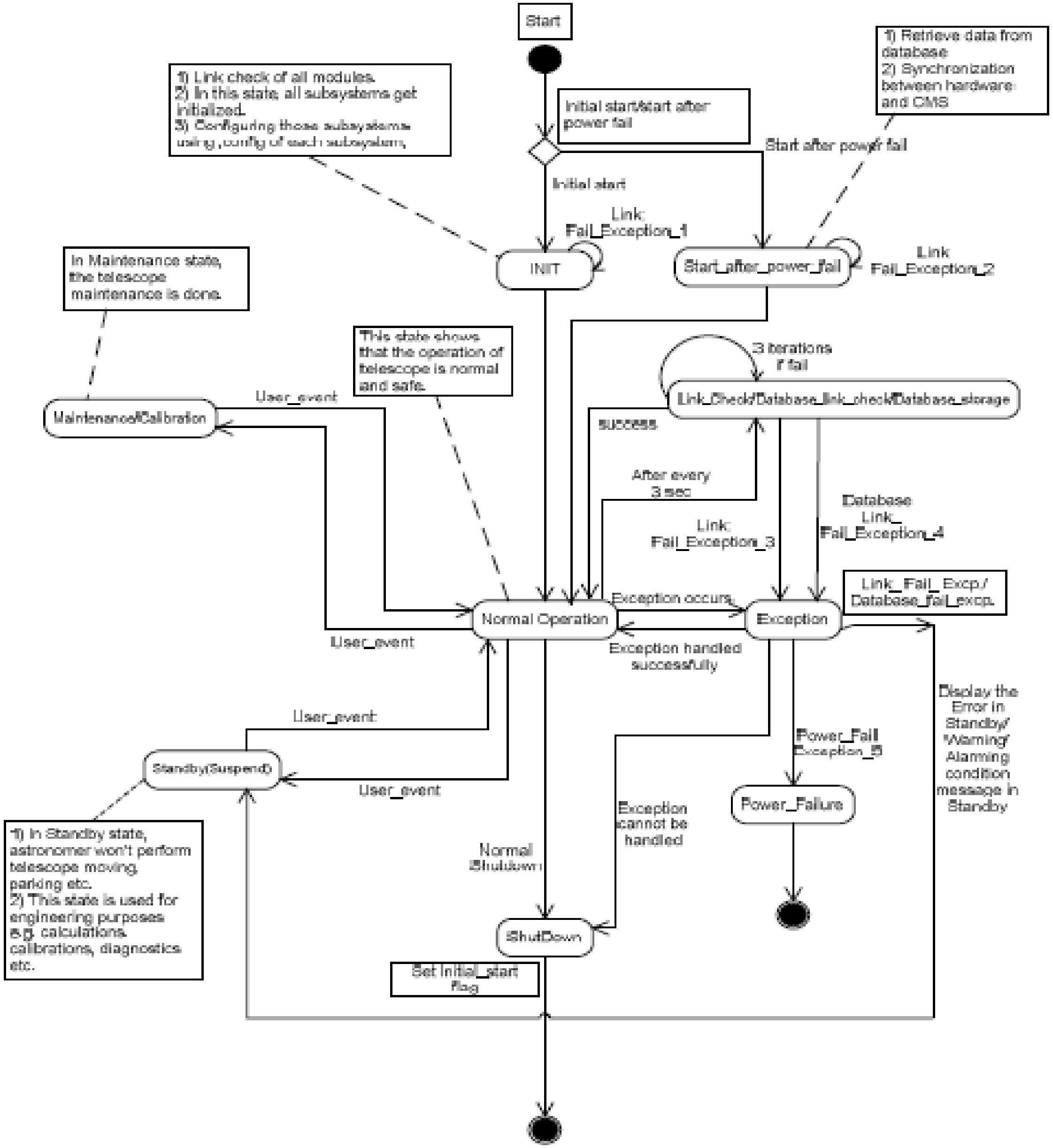
Components and Features

PERSISTENT
Partners in Innovation

State Machine Components

- States
- Transitions
- Actions
- Rules
- Alarms
- Monitoring Parameters

PERSISTENT



Start

1) Retrieve data from database
2) Synchronization between hardware and CMS

Initial start/start after power fail

1) Link check of all modules.
2) In this state, all subsystems get initialized.
3) Configuring those subsystems using ,config of each subsystem,

Start after power fail

Initial start

Link: Fail_Exception_1

INIT

Start_after_power_fail

Link Fail_Exception_2

In Maintenance state, the telescope maintenance is done.

This state shows that the operation of telescope is normal and safe.

3 iterations if fail

Link_Check/Database_link_check/Database_storage

User_event

Maintenance/Calibration

success

After every 3 sec

Database Link_ Fail_Exception_4

Link: Fail_Exception_3

Link_ Fail_ Excp./ Database_fail_excp.

Exception occurs

Normal Operation

Exception

User_event

Exception handled successfully

Display the Error in Standby/ Warning/ Alarming condition message in Standby

User_event

Power_Fail Exception_5

Standby(Suspend)

User_event

Power_Failure

Exception cannot be handled

1) In Standby state, astronomer won't perform telescope moving, parking etc.
2) This state is used for engineering purposes e.g. calculations, calibrations, diagnostics etc.

Normal Shutdown

ShutDown

Set Initial_start flag

States

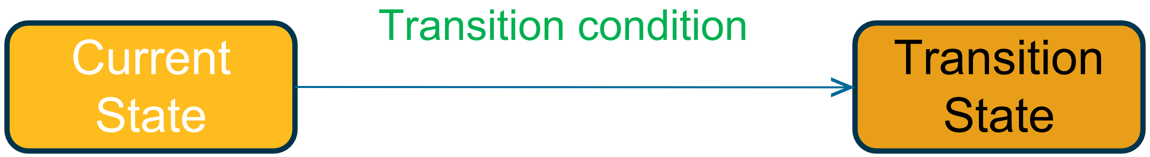
- CMS will have the following states:
- START – this state is the starting point for state-machine
- INIT – CMS will enter into init state from START state. In this state CMS will do connectivity check for all subsystems. On successful connections CMS will run an init batch script. This script will include the initialization commands. On successful execution of init script, CMS will move Normal State.
- NORMAL – In this state the normal CMS operations can proceed.
- DRIFT – This is an NCRA specific state where the servo subsystem would be assumed to be down and no command would be allowed to be sent to SERVO. The Engineer/Expert would have to manually change the CMS state to DRIFT mode.

States cont..

- MAINTENANCE - This state can be achieved through user driven operation in CMS.
- STANDBY – This state can be achieved through user driven operation in CMS.
- EXCEPTION – CMS will enter this state when there is an exception.
- INIT_ON_POWERFAILURE – CMS will enter this state from START state when the CMS comes up after power failure or when CMS has not been shutdown properly
- SHUTDOWN – This state can be achieved through user driven operation in CMS.

Transitions

- Transitions indicate the steps involved for moving from one state to another.
- To transition from one state to another the condition to enter the particular state needs to be defined. State transition occurs upon positive evaluation of condition.



Actions

- Actions are tasks done on entering a state.
- When CMS enters a state it can perform some actions associated with that state. For e.g. in INIT state the initialization scripts need to be executed.
- When a rule condition evaluates to true, the corresponding action specified by the rule will be executed. The details of conditions and actions supported in Rules-engine are specified in the Rules section.

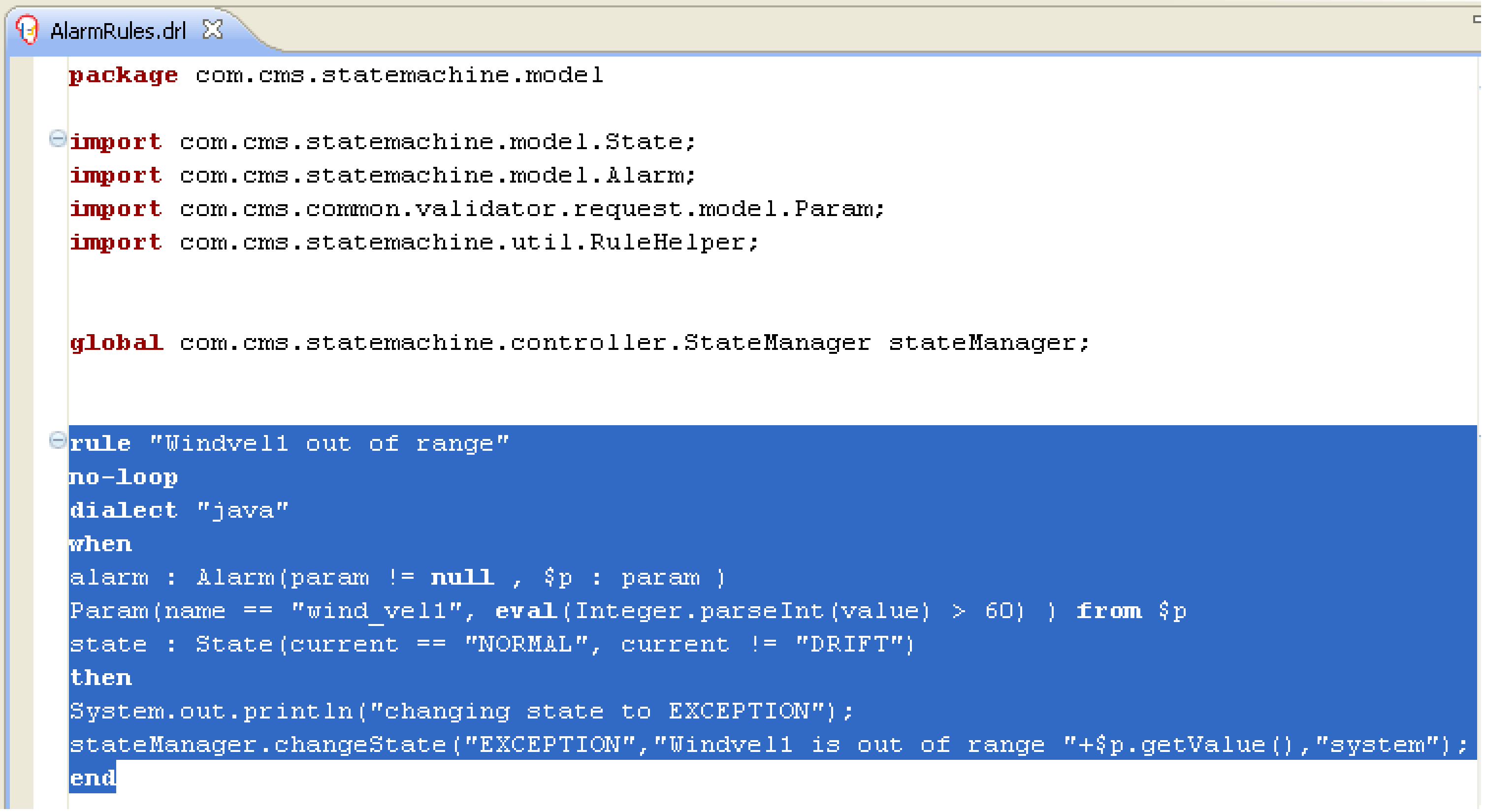
Rules

- Rules define the condition and the actions to be taken on particular condition.

- The conditions can be one or more of the following:
  - Alarm raised
  - Monitoring parameter goes out of range
  - CMS is in a particular state
  - Command fails
  - Response parameter out of range

- The actions can be one or more of the following:
  - Change state
  - Block some or all Commands to particular subsystem
  - Run a pre-defined batch script
  - Raise an Alarm

Sample Rule specified using JBoss Drools



```
package com.cms.statemachine.model

import com.cms.statemachine.model.State;
import com.cms.statemachine.model.Alarm;
import com.cms.common.validator.request.model.Param;
import com.cms.statemachine.util.RuleHelper;


global com.cms.statemachine.controller.StateManager stateManager;


rule "Windvel1 out of range"
no-loop
dialect "java"
when
alarm : Alarm(param != null , $p : param )
Param(name == "wind_vel1", eval(Integer.parseInt(value) > 60) ) from $p
state : State(current == "NORMAL", current != "DRIFT")
then
System.out.println("changing state to EXCEPTION");
stateManager.changeState("EXCEPTION","Windvel1 is out of range "+$p.getValue(),"system");
end
```

Alarms

- Alarms can be raised by subsystem by sending it as part of Response
- Alarm is raised when a monitoring param goes out of range.
- The Alarm associated with a monitoring param is specified in the <supportedalarms> section of the commands xml. The syntax to be followed for alarm name is alarm_<monitoring param name> . For e.g. the name for wind_vel1 parameter the alarm name would be alarm_wind_vel1

Monitoring Parameters

- Each subsystem will send monitoring parameters at specific interval. If the monitoring parameter is not received at specific interval of time and if the monitoring is not disabled then an alarm is raised and the subsystem will be assumed to be in exception state.
- The monitoring parameter will be validated against the validation specified in the <supportedresponses> section.

State Machine Features

- Monitor each monitoring parameter and raise alarm when parameter goes out of range. The range and alarm has to be pre-defined in CMS.
- Change state when an Alarm is raised. The rule for state change needs to be defined.
- Change to Exception state when an Critical Alarm is raised.
- Change State though UI
- Raise an Alarm for any persistent command failure. The rule for this will have to be defined in rules-engine.

State Machine Features cont…

- Take an action when an alarm is received. The actions will have to be specified through batch script. The Alarms will have to be pre-defined and the batch script to be called also needs to be pre-defined in the rules-engine.
- Raise alarm when the Antenna/Telescope goes off source after it was on source. This will be determined if the Altitude or Azimuth or both are consistently out of the allowed limits after being on source.
- Block commands (one or many) to a subsystem when state machine is in particular state. This rule has to be defined as to which commands need to be blocked.

State Machine Challenges and Limitations

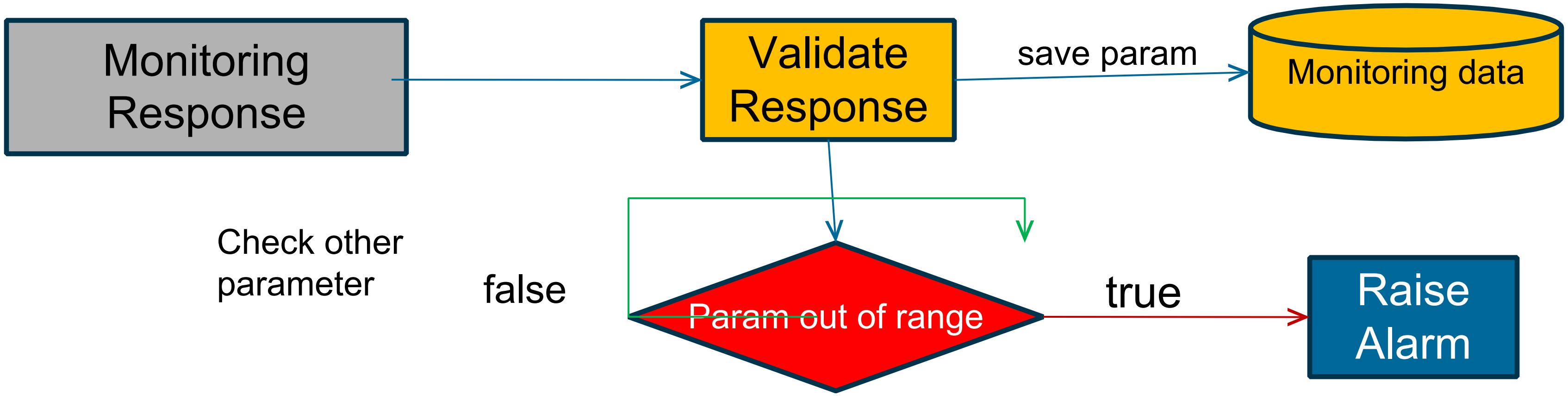
- States are not configurable, since the transitions and actions associated with a State requires coding. This implies that States, transitions and actions associated with it cannot be added through XML.
- The Rules are defined using JBoss Drools, so the Expert/Engineer need to get acquainted with the syntax of Drools http://www.jboss.org/drools/drools-expert.html
- State, Transitions and Actions need to be pre-configured.
- Rules have to be pre-configured and a tomcat restart would be required for the modified rules to get applied.

State Machine Challenges and Limitations cont...

- The actions mentioned in rules are limited to the functionality supported by state machine. The supported actions are as mentioned below:
  - Change state
  - Block some or all Commands to particular subsystem
  - Run a pre-defined batch script
  - Raise an Alarm

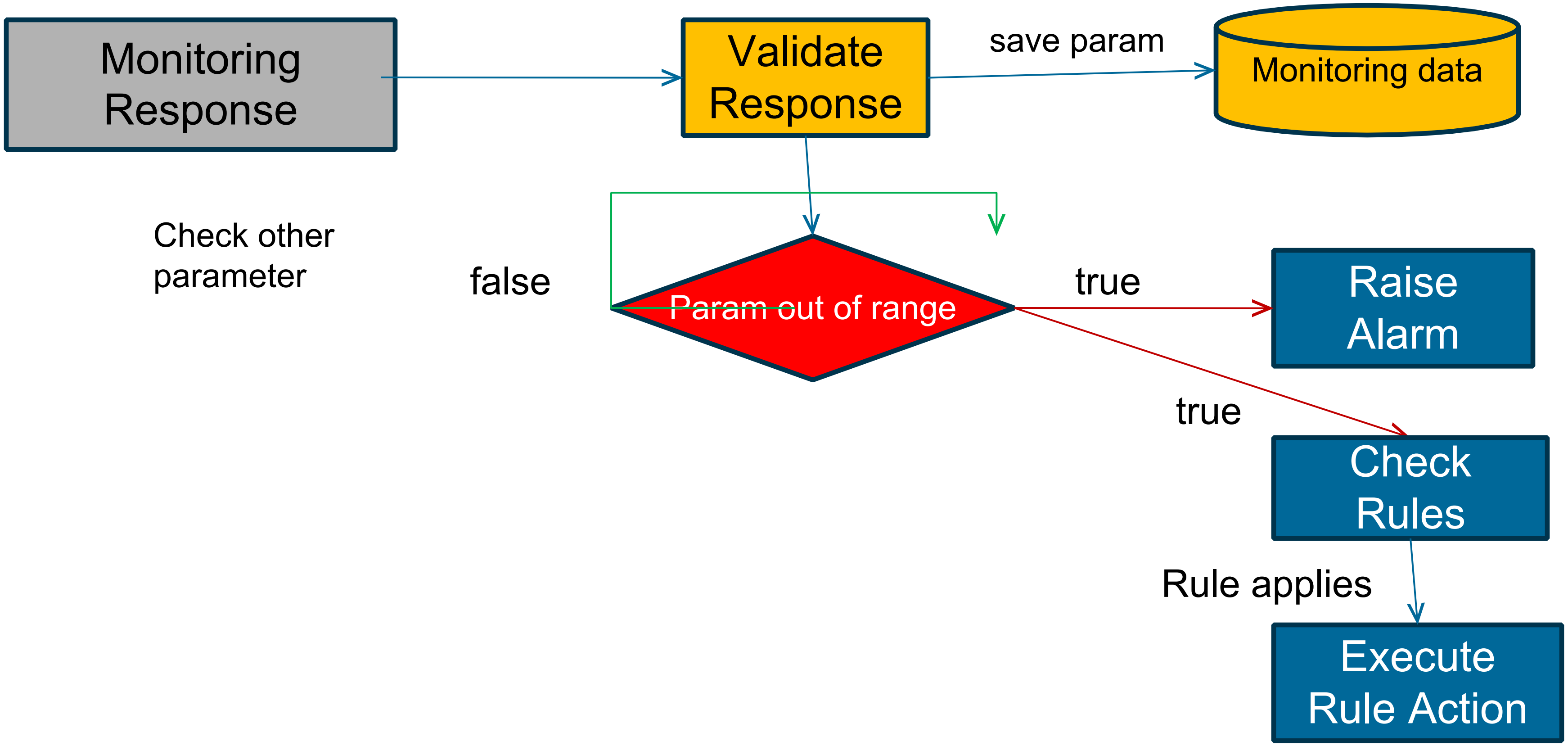
State Machine Demo – Scenario 1

- Raise alarm when monitoring parameter out of range



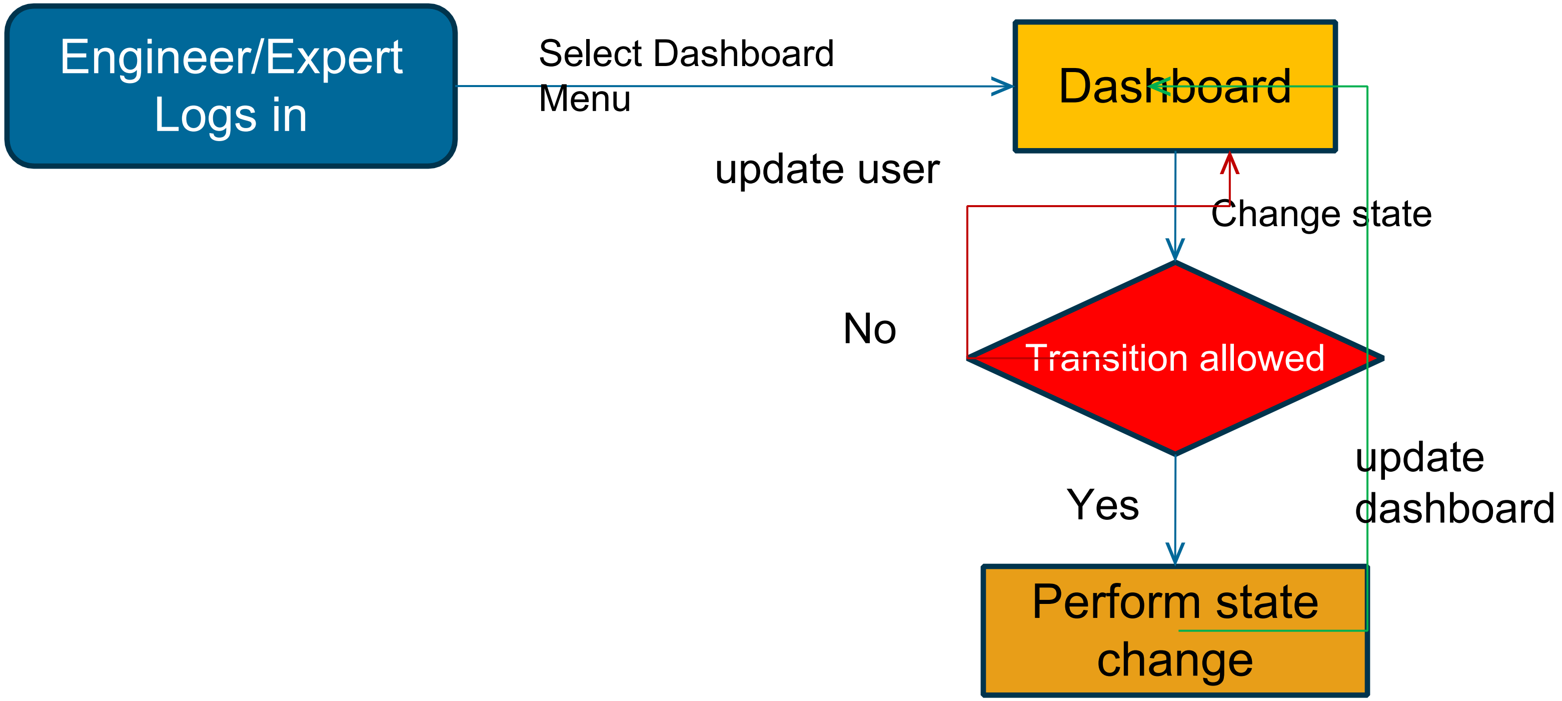
State Machine Demo – Scenario 2

- Apply rule when monitoring parameter is out of range



State Machine Demo – Scenario 3

- Change State through CMS UI



Inputs for State Machine :-

Requirement –

- Raising the alarm and executing the Exception handler routine if a particular parameter is out of the defined range – **Monitoring parameters** from wrapper

Inputs needed –

1. List of Alarms to be raised for the Monitoring parameters out of range
2. Exceptional handler for an alarm (Command to be sent from CMS)
3. Any other actions to be included for e.g.: - Block certain commands

Sr. No	Monitoring parameters (name and type) - from wrapper	Valid Range	Alarm level to be raised	Exceptional handler commands	Additional Actions

Inputs for State Machine :-

Requirement –

- Raising the alarm and executing the Exception handler routine for a persistent **Command failure** and/or a **parameter in Response is out of range**

Inputs needed –

1. List of Alarms to be raised for a persistent command failure and for parameter in Response out of range
2. Exceptional handler for above scenario (Command to send from CMS)
3. Any other action to be included for e.g. Block certain commands

Sr. No	Response parameter (name & type) - from wrapper / Command failure	Valid Range (Response Parameter)	Alarm (name, level) to be raised	Exceptional handler commands	Additional Actions

Inputs for State Machines :-

Requirement –

- **Init State Actions to be executed by CMS**
  - check if all wrappers are up
  - send init or any other command to subsystem

Inputs needed –

1. The List of activities to be performed after entering INIT state
2. The List of activities to be performed after entering SHUTDOWN state
3. The List of activities to be performed after entering MAINTENANCE state
4. The List of activities to be performed after entering any other state.

THANK YOU!