Change Log: Monitor Subsystem

Context (Object)

Added context object representing any of the ports that leave or enter through the edge of the context of the Monitor Subsystem.

Ports Represented:

- In
    - current_tempWstatus
    - lower_alarm_temp
    - upper_alarm_temp
- Out
    - alarm_control
    - monitor_status

Code

```
object Context {
 // ----- Inputs -------
 var current_tempWstatus_CONTEXT_In: TempWstatus_impl = TempWstatus_impl(z"85",
ValueStatus.Valid)

 var lower_alarm_tempWstatus_CONTEXT_In: TempWstatus_impl =
TempWstatus_impl(z"70", ValueStatus.Valid)
 var upper_alarm_tempWstatus_CONTEXT_In: TempWstatus_impl =
TempWstatus_impl(z"90", ValueStatus.Valid)

 // Operator settings (alarm upper and lower temperature)

 // ------ Outputs ------

 var alarm_control_CONTEXT_Out: On_Off.Type = On_Off.Off

 var monitor_status_CONTEXT_Out: Status.Type = Status.Init_Status

}
```

App Object

The app object was created and is just used to wrap any of the previous elements' ports, initialize, and computational functions.

Manage Monitor Interface's Section Changes:

- initialise_EP_TS() changed to initialise_MMI
- firstInvocationFlag changed to firstInvocationFlag_MMI
- lastMonitorMode changed to lastMonitorMode_MMI
- timeout_condition_satisfied to timeout_condition_satisfied_MMI

Mange Monitor Mode's Section Changes:

- initialise_EP_TS() changed to initialise_MMM()
- firstInvocationFlag changed to firstInvocationFlag_MMM
- lastMonitorMode changed to lastMonitorMode_MMM
- timeout_condition_satisfied to timeout_condition_satisfied_MMM

Mange Alarm Mode's Section Changes:

- initialise_EP_TS() changed to initialise_MA()
- lastCmd changed to lastCmd_MA
- firstInvocationFlag to firstInvocationFlag_MA

Comm Object

Communication object added in similar manner to the one found in Regulator's abstract implementation. It should be noted that Hatcliff's comments on the Regulator abstractions Comm object mentions that something needs to be done with the interface_failure_IN value that needs to be populated; the Comm object in Monitor subsequently an outgoing interfact_failure_OUT value into context, and has a similar population vacancy for its internal value which Regulator's subsystem could theoretically provide.

It should also be noted that the Monitor abstract Comm object was complaining about a missing modifies clause, which I have added, but it is still a difference between the two.

```
object Comm {

 // ------- C o n t e x t   to   M M I
 def CONTEXT_to_MMI_current_tempWstatus(): Unit = {
   Contract(
```

```
    Modifies(
      App.current_tempWstatus_MMI_In
    )
  )
  App.current_tempWstatus_MMI_In = Context.current_tempWstatus_CONTEXT_In
}

def CONTEXT_to_MMI_lower_desired_tempWstatus(): Unit = {
  Contract(
    Modifies(
      App.lower_alarm_temp_MMI_Out
    )
  )
  App.lower_alarm_temp_MMI_Out = Context.lower_alarm_tempWstatus_CONTEXT_In
//TODO
}

def CONTEXT_to_MMI_upper_desired_tempWstatus(): Unit = {
  Contract(
    Modifies(
      App.upper_alarm_temp_MMI_Out
    )
  )
  App.upper_alarm_temp_MMI_Out = Context.upper_alarm_tempWstatus_CONTEXT_In
}

// ------- M M I  to C o n t e x t

def MMI_to_CONTEXT_monitor_status(): Unit = {
  Contract(
    Modifies(
      Context.monitor_status_CONTEXT_Out
    )
  )
  Context.monitor_status_CONTEXT_Out = App.monitor_status_MMI_Out
}

// ------- M M I  to   M A

def MMI_to_MA_lower_desired_temp(): Unit = {
  Contract(
    Modifies(
       App.lower_alarm_temp_MA_In
    )
  )
  App.lower_alarm_temp_MA_In = App.lower_alarm_temp_MMI_Out
}

def MMI_to_MA_upper_desired_temp(): Unit = {
```

```
  Contract(
    Modifies(
      App.upper_alarm_temp_MA_In
    )
  )
  App.upper_alarm_temp_MA_In = App.upper_alarm_temp_MMI_Out
}

// ------- M M I to M M M

def MMI_to_MMM_interface_failure(): Unit = {
  Contract(
    Modifies(
      App.interface_failure_MMM_In
    )
  )
  App.interface_failure_MMM_In = App.interface_failure_MMI_Out
}

// ------- C o n t e x t  to  M M M

def CONTEXT_to_MMM_current_tempWstatus(): Unit = {
  Contract(
    Modifies(
      App.current_tempWstatus_MMM_In
    )
  )
  App.current_tempWstatus_MMM_In = Context.current_tempWstatus_CONTEXT_In
}

def CONTEXT_to_MMM_monitor_failure(): Unit = {
//TODO Detect Monitor Failure
}

// ------- M M M  to  M M I

def MMM_to_MMI_monitor_mode(): Unit = {
  Contract(
    Modifies(
      App.monitor_mode_MMI_In
    )
  )
  App.monitor_mode_MMI_In = App.monitor_mode_MMM_Out
}

// ------- M M M  to  M A

def MMM_to_MA_monitor_mode(): Unit = {
  Contract(
```

```
    Modifies(
      App.monitor_mode_MA_In
    )
  )
  App.monitor_mode_MA_In = App.monitor_mode_MMM_Out
}

// ------- C o n t e x t  to  M A

def CONTEXT_to_MA_current_tempWstatus(): Unit = {
  Contract(
    Modifies(
      App.current_tempWstatus_MA_In
    )
  )
  App.current_tempWstatus_MA_In = Context.current_tempWstatus_CONTEXT_In
}

// ------- M A  to  C o n t e x t

def MRM_to_MHS_regulator_mode(): Unit = {
  Contract(
    Modifies(
      Context.alarm_control_CONTEXT_Out
    )
  )
  Context.alarm_control_CONTEXT_Out = App.alarm_control_MA_Out
}

// TODO:  Dealing with internal failure (MMM):: (Regulator has interface
failure missing, this the other, probably need comm, between two.)
}
```