OBD Fault Manager Introduction

Woodward

Engine Systems

Outline

- **Existing MotoHawk Fault Manager**
  - Strengths/Weaknesses
- **Introduction to the OBD Fault Manager**
  - Concepts
  - Blocks
- **Questions**

Fault Manager Highlights

- Distributed fault detection
- Marquees display the current states
- 3 fault states with 6 modes
- User has blocks to iterate through faults
- Fault actions decouple detection logic from response logic

What is “OBD Compliant”?

- **The details depend on where the application will be deployed**
  - Most countries have active or pending regulations
  - Regulation “sharing”
  - Protocol support impacts

- **OBD compliance is a long and arduous task**
  - Emissions regulations
  - Separation of normal vs. emissions-related faults
  - Bureaucratic governments
  - Regulation interpretation
  - High quality and well tested diagnostics
  - Failed component testing
  - Documentation

Fault Manager Limitations

- **No concept of “Drive Cycle” which is critical for bookkeeping purposes**
- **All faults have the same category. There is no distinction between emissions-related faults and normal faults.**
- **Fault related data is difficult to implement in Simulink because there is no built in support**
  - Protocol integration is difficult
- **Existing fault states do not correspond to legislated definitions of required fault states (suspected, active, occurred)**
- **No concept of “Permanent” faults**

Introducing the OBD Fault Manager

- **What is the OBD Fault Manager?**
  - It is a **new** Fault Manager designed to accommodate OBD compliant systems.
  - Intended to be flexible enough to satisfy multiple OBD standards
  - Intended to function elegantly with many different types of protocol handlers

- **What does it do?**
  - It contains 12 states that track the progress of a fault from initial detection to fault storage to clearing.
  - Introduces new features for fault action routing that are not present in the Fault Manager.
  - Introduces the concept of “Fault Related Data”
  - Introduces the concept of drive cycles

- **What does it not do?**
  - By itself, it does not satisfy all OBD requirements. An OBD infrastructure still needs to built in the application.

OBD Fault Manager Concepts

- **Fault States**
- **Fault Action Routing**
- **Fault-related Data**

OBD Fault Manager Concept – Fault States

- **Fault States – The ability to map a particular unit of data (e.g. Occurrence Counts) to a particular fault in the application.**

- **Fault Manager:**
  - 3 states
    - Suspected
    - Active
    - Occurred
  - 6 modes
    - Disabled
    - Enabled
    - Sticky
    - Enabled Persistent
    - Sticky Persistent
    - Save Occurred
- **OBD Fault Manager:**
  - 12 states (described in the following slides)
  - Protocol features are enabled (e.g. Readiness, Previously Active, 14229-1 states, etc.)

OBD Fault States

Name	Storage Type	Clear Condition	Description
Suspected	Volatile	Fault definition input is low	This is the initial observation of fault behavior. It is prone to "noise" and is thus put through a X of Y filter to determine whether the fault is Pending or not. Both the X and Y counters are reset if the X limit or the Y limit is exceeded.
Pending	NonVolatile	Mark To Clear block parameter is set to "Pending "and the inhibit input is 0 at least once during a drive cycle. The state is deactivated at the end of the drive cycle. A fault clear will also deactivate this state.	If the fault logic determines that a fault is pending (i.e. it was detected at least X out of Y samples), then the Pending state is activated. This will be known as the **pending filter**.
Confirmed	NonVolatile	Mark To Clear block parameter is set to Confirmed and the inhibit input is 0 at least once during a drive cycle. The state is deactivated at the end of the drive cycle. A fault clear will also deactivate this state.	A fault is considered confirmed if it has passed the pending filter mentioned above in X of Y *completed* drive cycles. The default value for each fault is set via the Drive Cycles and Total Drive Cycles parameters. The Drive Cycles and Total Drive Cycles can be set to 0 if a fault should be confirmed immediately.

OBD Fault States (cont'd)

Ready	NonVolatile	Mark To Clear block parameter is set to Ready and the inhibit input is 0 at least once during a drive cycle. The state is deactivated at the end of the drive cycle. A fault clear will also deactivate this state.	The ready status denotes that the pending filter has been executed to completion at least once. This state latches that value.
Failed This Drive Cycle	Volatile	Clears at the end of the drive cycle	Failed This Drive Cycle latches a fault detection from the pending filter once per drive cycle. It is present to ensure that the Drive Cycle counts only increment once in a given drive cycle.
Test Complete This Drive Cycle	Volatile	Clears at the end of the drive cycle	This is similar to Ready, but is not latched in NonVolatile. If a fault detection has executed to completion, then this state is activated.

OBD Fault States (cont'd)

MIL Request	NonVolatile	Mark To Clear block parameter is set to MIL Request and the inhibit input is 0 at least once during a drive cycle. The state is deactivated at the end of the drive cycle. A fault clear will also deactivate this state.	MIL Request is intended to assist the user with illuminating the MIL light in their application. The fault manager does not attempt to control an output in the module. Instead, it notifies the user when a fault should be turning on the MIL. If a fault is "emissions-related" and activates the confirmed state, then the MIL Request state will also be activated. It is the responsibility of the user to manage the state.
Previously Active	NonVolatile	Mark To Clear block parameter is set to Previously Active and the inhibit input is 0 at least once during a drive cycle. The state is deactivated at the end of the drive cycle. A fault clear will also deactivate this state.	Active and Confirmed are interchangeable within the OBD fault Manager. The Previously Active state informs the user whether a fault had an active confirmed state and transitioned to inactive at some point. It latches this value.
Permanent	NonVolatile	Mark To Clear block parameter is set to Permanent and the inhibit input is 0 at least once during a drive cycle. The state is deactivated at the end of the drive cycle. A fault clear will **NOT** deactivate this state.	The permanent state is similar to Confirmed except that the state is only activated if it is calibrated as a Permanent fault.

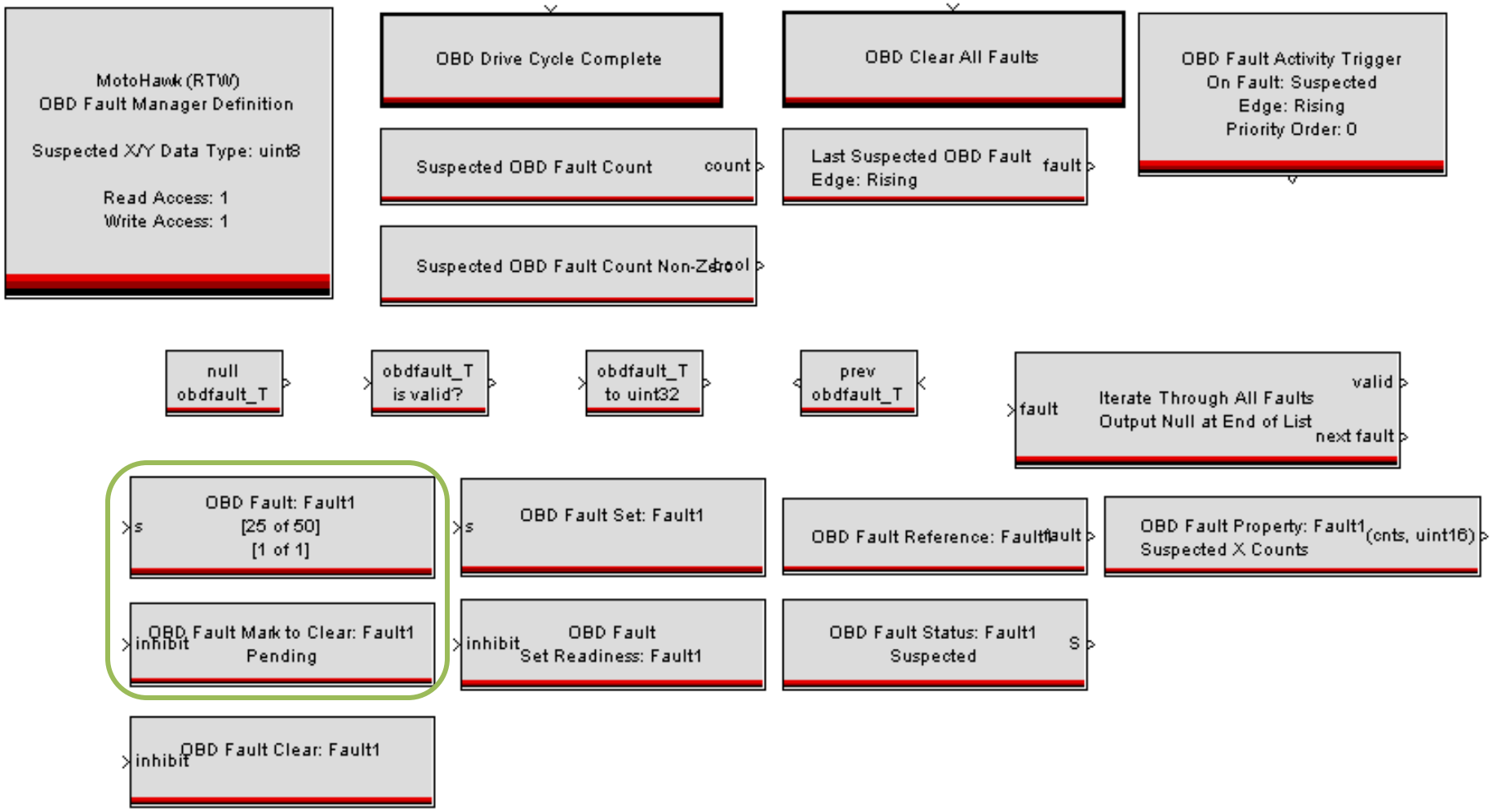
OBD Fault States (cont'd)

Test Failing	Volatile	A detection cycle that does not detect the fault must occur.	This state reports the most recent Suspected X/Y detection results. It is similar to Pending except the state does not latch in the "active" state.
Test Failed Since Last Clear	NonVolatile	Fault clear will deactivate this state.	This state is active if a fault has failed since a fault clear command occurred.
Failed Last Drive Cycle	NonVolatile	Clears at the end of the current drive cycle if a fault did not occur	This state reports whether the fault failed on the previous drive cycle.

OBD Fault States

- NonVolatile fault states do not “self heal”
- Intervention by the application is necessary
- This is called “bookkeeping” and usually occurs at the end of a drive cycle

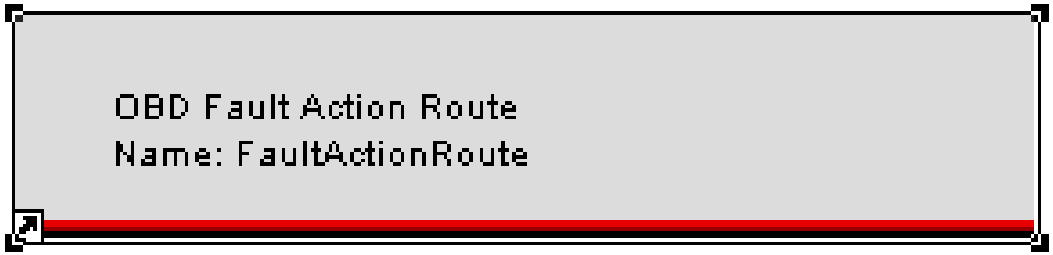
OBD Fault States



OBD Fault Manager Concept – Fault Action Routing

- **Fault Action Routing links fault state(s) to recovery or other types of logic in the application**
  - Examples : Limp home, MIL illumination
- **Fault Manager:**
  - Fixed number of action routes (4)
  - All routes can be calibrated
- **OBD Fault Manager:**
  - Adjustable number of action routes
  - Routes can be fixed (no calibrations)

OBD Fault Action Routing



OBD Fault Action Route Defaults

Fault Action Route Contents:

Fault Action Route Name	Fault Name	Fault Action	Fault Condition
FaultActionRoute	Fault1	MyFaultAction	Suspected

OBD Fault Manager Concept – Fault Data

- **Fault Data – The ability to map a particular unit of data (e.g. Occurrence Counts) to a particular fault in the application.**

- **Fault Manager:**
  - No built in mechanism

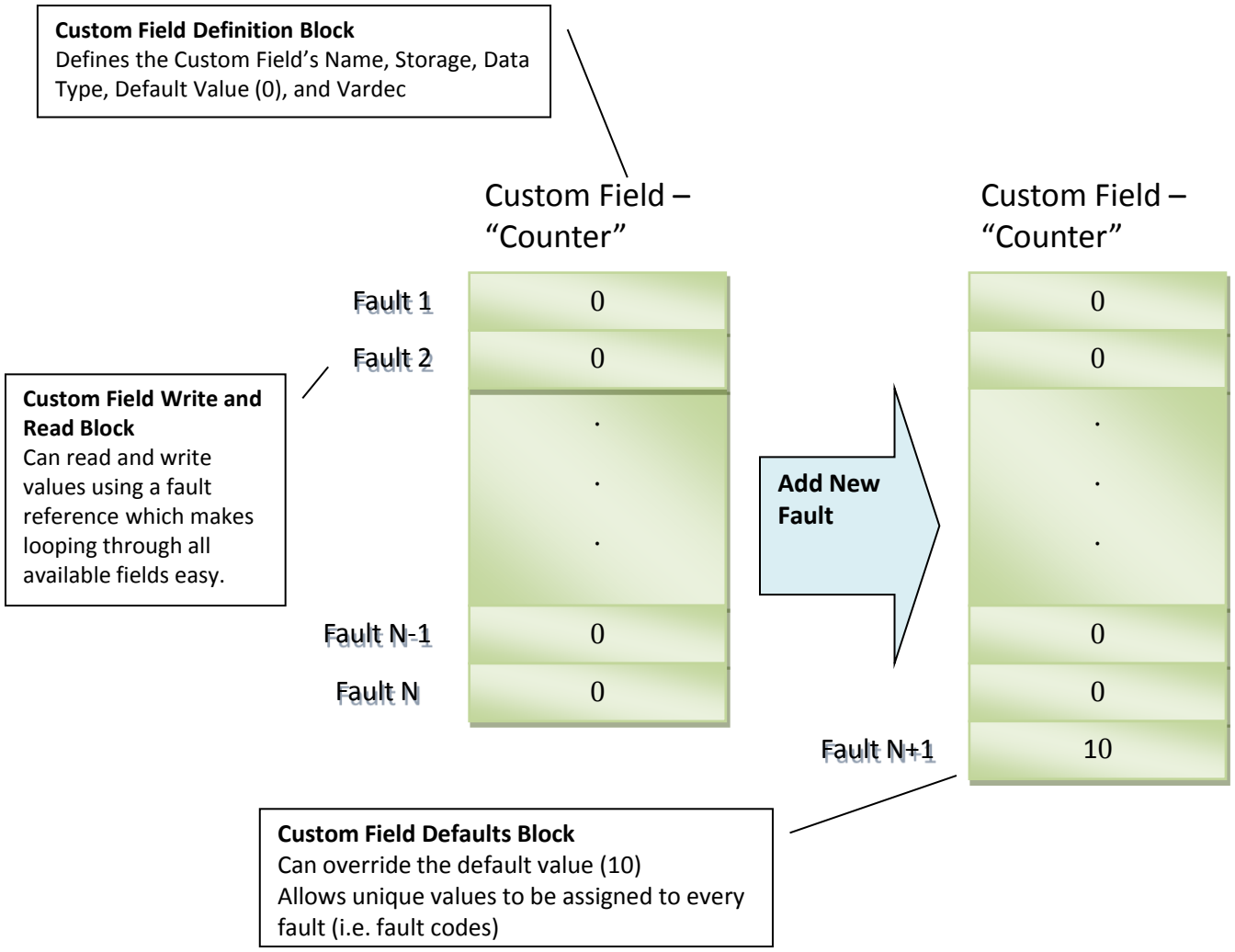
- **OBD Fault Manager:**
  - "Custom Fields" track this data
    - Dynamically sized and ordered at build time by MotoHawk
    - Can reference faults via the "obdfault_T" data type in Simulink
  - Protocol features are enabled (e.g. DTC, SPN, FMI, Occurrence Count, etc.)

Custom Field Examples

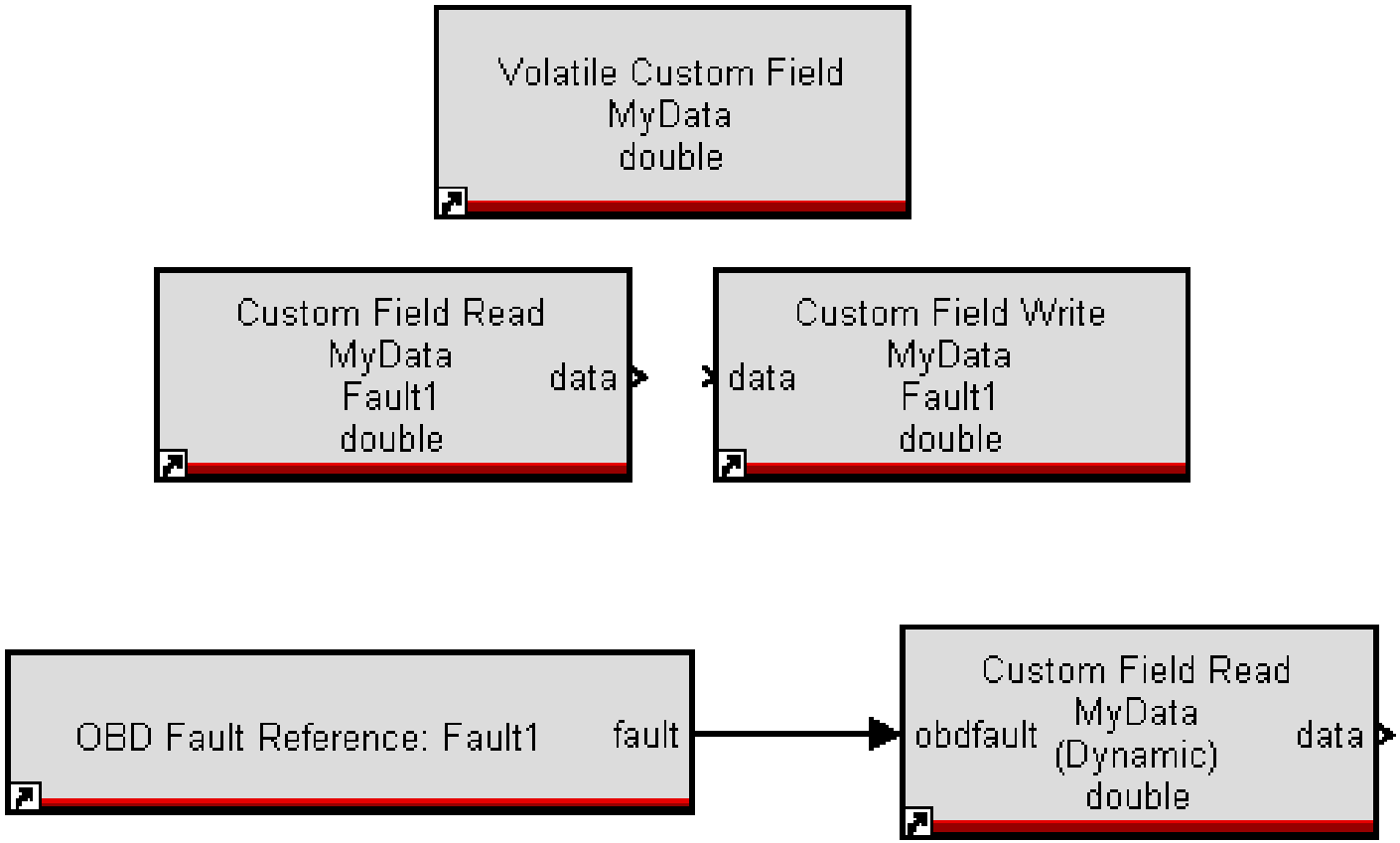
- **Required for All OBD Faults**
  - SPN (uint32), Constant – DM1, 2, 6, 12, 23, etc
  - FMI (uint8), Constant – DM1, 2, 6, 12, 23, etc
  - Occurrence Count (uint8), Non-volatile – DM1, 2, 6, 12, 23, etc
  - Fault Classification (uint8), Calibration – DM41-52
  - MIL Counts(uint8), Non-volatile – for turning off MIL.
  - Disable (boolean) – for Enable/Disable OBD Fault
  - ConsecutiveNonFailCycles (uint8) – for clearing Pending Fault

- **Optional for other specific Faults**
  - Readiness (uint8), non-volatile – DM5
  - Total Active Time (uint16) non-volatile– DM32
  - Total Previously active Time (uint16), non-volatile – DM32
  - DTC Time until Derate (uint16), non-volatile – DM32 (Derate fault only)
  - Failure Specific B1 Count (uint16), non-volatile – DM40 (EOBD VI B1 Classification only)

Custom Field Details



Custom Data Field



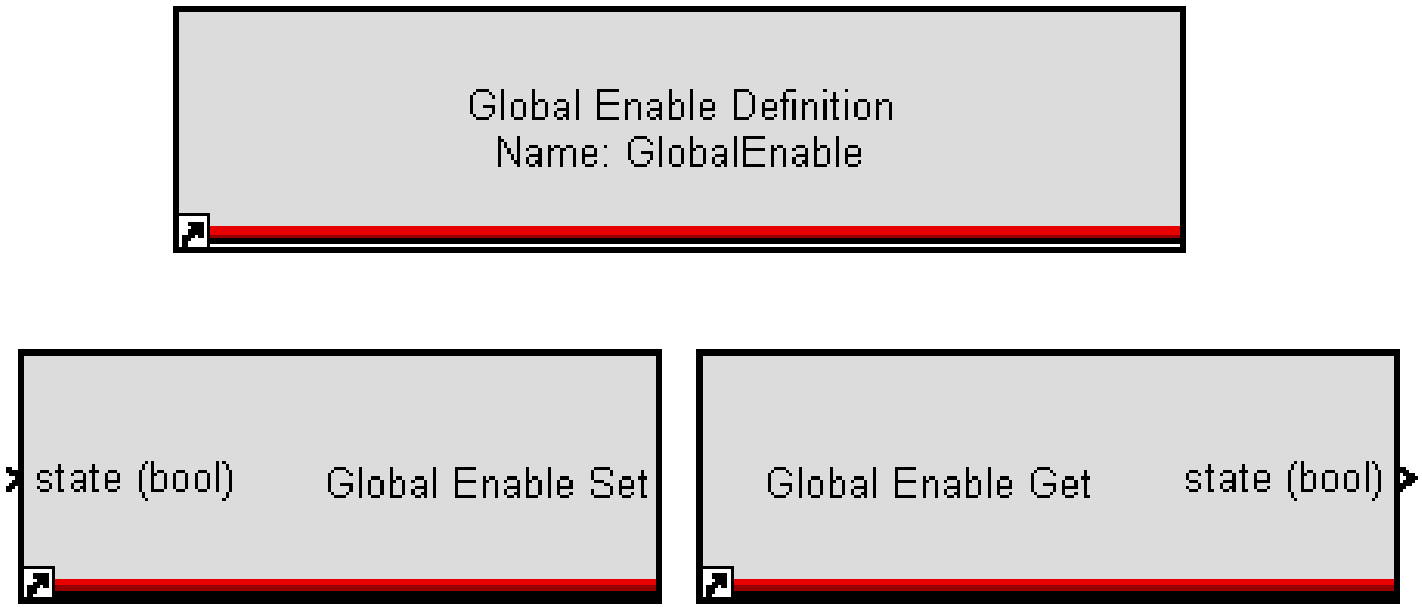
OBD Fault Manager Concept – Global Disable

- **Global disable – the ability to “freeze” faults (prevent new faults from being set)**

- **Fault Manager:**
  - No built in mechanism

- **OBD Fault Manager:**
  - Global disable block will prevent all faults from being set
  - Protocol features are enabled (e.g. Service 0x85 in 14229-1)

Global Enable



OBD Fault Manager Concept – IUPM

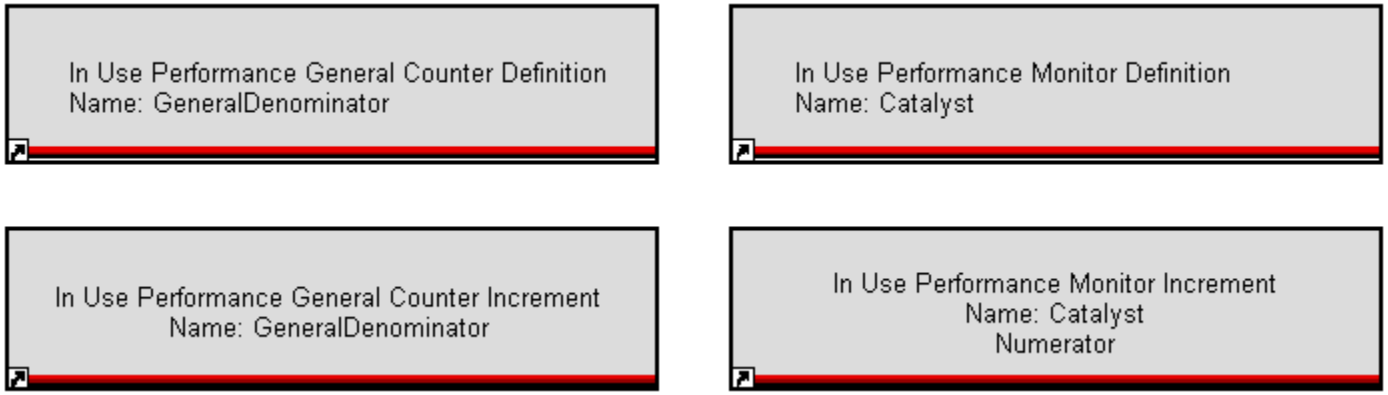
- **In Use Performance Monitor Ratio – This is an OBD concept to track whether a given diagnostic is run often enough to satisfy regulation requirements.**

- **Fault Manager:**
  - No built in mechanism

- **OBD Fault Manager:**
  - Drive cycle is required
  - In Use Performance Monitor blocks manage the counts according to OBD regulations
  - Protocol features are enabled (e.g. DM20 in J1939-73)

Performance Counters

- **Ratio of Tests Run**
- **General Counters**



Future Work

- **Simulation support**
- **OBD Automatic Documentation**
- **Further Communication Integration**
  - J1939
  - J1979
  - 14229-1
  - ISO27145
  - Others?