The Master Monitor for the Green Bank Telescope’s Active Surface

Amy L. Shelton
Digital Engineer
Green Bank, West Virginia

Initial design and coding done by Richard J. Lacasse
Active Surface Hardware
Active Surface Software

- Bus Granter Task
- LVDT Temperature Task
- Actuator Time Logger Task
- RPC Task
- Dispatch Task
- Master Monitor Task
How Does the Master Monitor Fit in to the GBT Project?

- The Green Bank Telescope
- Active Surface
- Active Surface Monitoring Points
  - Actuators, IIOPs, Control Modules, Master Oscillator, SIB, RPC, Power Supplies, Airflow in the Actuator Room, Actuator Room Temperature, Tool Box in the Actuator Room, Watchdogs, Emergency Stop, Emergency Stop Bypass
How Does the Master Monitor Work?

- Gathers Information
- Updates State Transition Diagrams
  - Examines gathered information and determines the state transition
  - Updates the current state, generates appropriate messages and sets “to do” list flags
- Updates the Global Status and the Actuator Status
  - Master Monitor Vs. Global/Actuator Status
A state transition diagram is a pictorial representation of a finite state machine. Usually, the states are represented as circles and the transitions are represented as directed line segments.

A finite state machine is a function which maps an ordered sequence of input events into a corresponding sequence of (sets of) output events*.

* [http://wombat.doc.ic.ac.uk/foldoc/foldoc.cgi?finite+state+machine](http://wombat.doc.ic.ac.uk/foldoc/foldoc.cgi?finite+state+machine)
Example of a State Diagram

Name: \texttt{toolStowTest}
Run Conditions: \texttt{sibTestState = OK}
\textit{ERR} = \texttt{toolstow bit \neq 0}
\textit{OK} = \texttt{toolstow bit = 0}

\begin{itemize}
  \item OK
  \item ERR
  \item WARN
  \item FAIL
\end{itemize}

- OK: \texttt{msgs off reset fault bit restore enables}
- ERR: \texttt{w msg}
- WARN: \texttt{f msg disable act motor ps off transnet ps off save enables set fault bit}
- FAIL: \texttt{f msg disable act motor ps off transnet ps off save enables set fault bit}

The diagram illustrates the state transitions and conditions for the state machine.
Another State Diagram

Name: tNetSupplyTest
Run Conditions: sibTestState = OK
ERRO = supply in warning range
ERR1 = supply in error range
ERR2 = supply in not-off range
OK = supply in spec
Note: 4 tests, one per supply
Implementation of a State Diagram in C++

- The Master Monitor uses switch statements to call appropriate methods.
  - Current state
  - Transitions
- Methods may execute one or more actions associated with the transition and update the current state.
void MasterMonitor::toolStowTest
{
    testValue = setToolStowTestValue();   // Criterion for switching st
    switch (toolStowTestState)  // State table implementation
    {
        case MasterMonitor::AS_OK:
            switch (testValue)
            {
                case MasterMonitor::ASV_C
                    break;
                case MasterMonitor::ASV_ERR
                    toolStow_OK_ERR();
                    break;
                default
                    defaultErrMsg();
                    break;
            }
            break;
        case MasterMonitor::AS_WARN:
            switch (testValue)
            {
                case MasterMonitor::ASV_C
                    toolStow_WARN_OK();
                    break;
                case MasterMonitor::ASV_ERR
                    toolStow_WARN_ERR();
                    break;
                default
                    defaultErrMsg();
                    break;
            }
            break;
        case MasterMonitor::AS_FAIL:
            switch (testValue)
            {
                case MasterMonitor::ASV_C
                    toolStow_FAIL_OK();
                    break;
                case MasterMonitor::ASV_ERR
                    break;
                default
                    defaultErrMsg();
                    break;
            }
            break;
        default:
            defaultErrMsg();
            break;
    }
}
Questions?