Controlled Execution of Application Timers from TTCN-3 Test Cases

Strategy and Benefits

From Motorola India
Bhaskar Rao G, Hyderabad
email: bhaskarraog@motorola.com
Satish G V K S & Keerthi Timmaraju, Bangalore

Agenda

- Application overview
- Problem description and issues
- Strategy with TTCN-3
- Test System architecture
- Technical details
- Summary

Application Overview

- Application Testing developed using
  - MDA (Model Driven Architecture)
  - UML 2.0
  - Telelogic Tau G2
- Model Testing using
  - Integrated Environment of Model Verifier of Tau G2 and TTCN-3

Problem & Solutions

- Handling Timer Sets inside the model
- Solution 1
  - Ignore the timer set
    - Timer Behavior will not be tested
    - Incomplete code-coverage can lead to bugs
- Solution 2
  - Allow the application to handle timer timeouts
    - Application has to wait out for the entire duration
      - Takes the control away from the test system
    - Application can immediately timeout
      - Resulting in errant behavior of the system

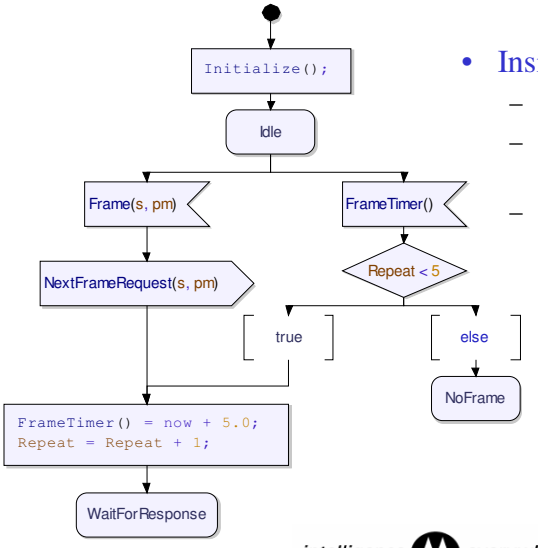
Results

- The solutions considered resulted in
  - Ineffectual testing of the application
  - Incomplete Code/Model Coverage
  - Manual Intervention reducing the automation efforts
  - Reducing the control of the test system over the application testing

TTCN-3 based Strategy

- Using the TTCN-3 test system to keep track of the timers set
  - When a timer is set in the model, test system is intimated of the timer set
  - Test system maintains a test component, esp. for the timer sets/resets/timeouts
  - This test component, based on the test case, resets or times out the timer
  - Test System in turn sends this as a message to the model

What happens…

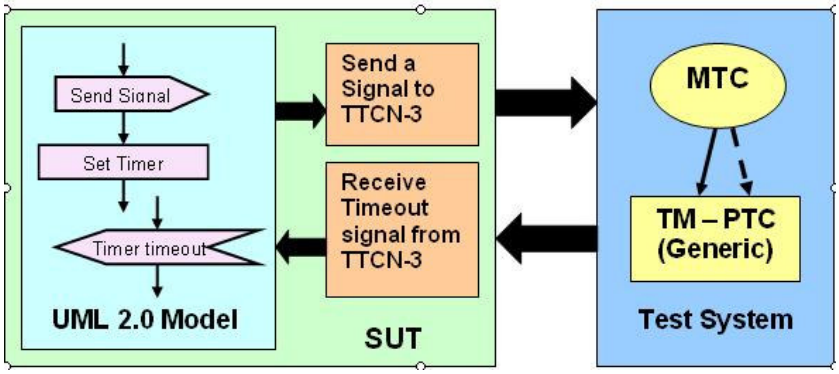


- Inside the model
  - Normal Timer Operation
  - Timer Sets according to the requirement
  - Wait for its expiry

What happens…

- Inside the Model-Environment glue code and Tau Kernel
  - Timer Set in model calls SDL_Set() in Tau Kernel
  - In SDL_Set,
    - Create a unique integer (timer_handle) for the timer
    - Store the timer and the time in a global data structure using the unique id
    - Send a TCP Message to the test case with timer_handle and timer value

Test System Architecture



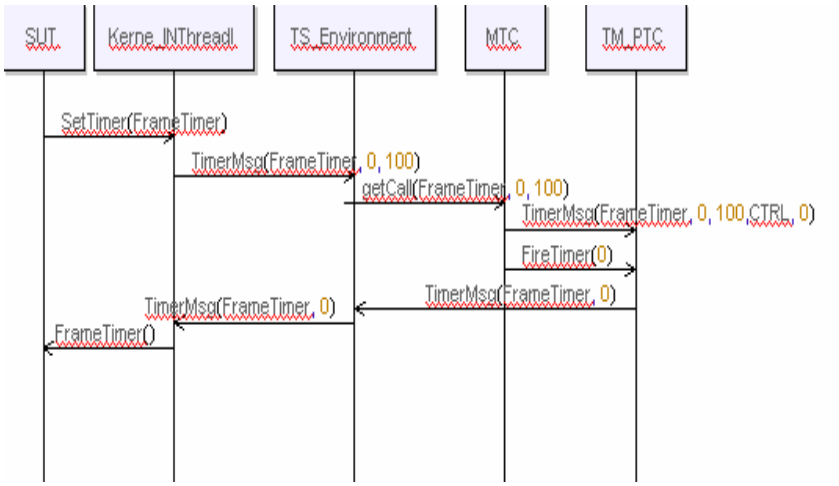
Inside the TC…

- A Parallel Test Component (PTC), TimerManager is created to handle timer events
- When expecting a timer set
  - The TC waits on getCall
- On receipt of the message
  - MTC hands the message to TimerManager
  - MTC control the timer behavior (expire immediately, expire only when demanded, expire as per the actual time)
  - TimerManager sets the timer
- When the timer expires
  - TimerManager sends the timer message with timer_handle to the glue code of the model

Inside glue code…

- xInEnv has a separate timer port to listen to
- When message is received on the timer port
  - Maps back to the timer based on Timer_Handle
  - Sends the appropriate timer signal to the application

Message Exchanges



Summary

- Controlled execution of timers enables greater model coverage during verification
  - Enhances quality
- The strategy can also be extended to integration testing and other controlled testing phases

Q & A