

TTCN-3 性能和实时测试例子工程

实时系统必须尊重时序特殊要求。功能经常关系到消息和过程调用的时机。因此,在这种情况下只检查消息值和消息顺序是不够的。测试组件必须能够检查消息是否被及时接收,并且必须能够控制刺激 stimulation 待测时机。

因此,测试语言必须提供测量时间的方法,定义时间点和时间跨度,以控制刺激的时机,并计算和比较时间值。此外,测试执行引擎必须确保在指定的动作(时间测量,定时刺激)按照要求的时间精度被正确执行。

为了充分满足测试实时系统的要求,定义了下面的 TTCN-3 核心语言扩展。

- 一个测试系统范围内可用的测试系统时钟,允许测试例执行过程中的时间测量。
- 直接且精确地访问测试系统与待测系统之间的有关交互事件的时间点。

1. 测试系统时钟

在 RT TTCN-3 时间进度是用测试系统时钟来度量。在每个测试例开始执行时时钟被初始化(设定为 0.0),在测试运行过程中每个组件都可以访问这个时钟。时钟值以浮点数来表示。系统时钟和 TTCN-3 已有的计时机制对时间进度同步。

1.1 访问当前的系统时间

测试系统时钟的当前值以“now”符号来表示,这个“now”符号被作为一种 TTCN-3 表达方式表示以秒为单位的当前测试系统时钟值。测试系统时钟值以浮点数来表示。Now 符号可以被应用到每个测试例定义以及 function 定义。但是不允许被用在 control part。

1.2 系统时间精度

测试系统时钟的整体精度的要求可以由步长注解的方式来指定。步长注释只允许用于模块,并且可以被用来说明由测试系统时钟提供了最小的必要时间测量的精度。

精度由一个十进制字符串值来定义,代表了由测试系统时钟描述的以秒为单位的最小的必要时间距离。一个具体的测试系统必须满足由步长注解给予足够的各自的测试用例定义执行的要求。当测试系统是不足够的测试用例执行的用户应被告知,至少试运行应与一个错误的判决结束。

2. 用于实时测量的通信端口类型

这个扩展包用 realtime clause 扩展基于消息和基于进程的端口类型定义。端口便于测试组件间以及测试组件与测试系统界面的通信。

Only instances of ports with a realtime clause shall be used for real-time measurements. This means, the redirection operator -> timestamp shall only be used by receiving operations (i.e. the operations receive, trigger, getcall, getreply and catch) applied to ports with a realtime clause.

3. 测量专用传入通信事件的定时信息

测试实时系统需要确切的直接关系到测试系统和待测系统通信(接收、分发消息以及程序调

用)的定时信息。

由 now 符号或 TTCN-3 的定时结构来获得定时信息和测试程序的逻辑结构相关,因此,它允许在 TTCN-3 语句级别的测量。TTCN-3 的语句级时间测量可能会受到阻塞队列,解码和匹配程序影响。这不是准确的测试系统和待测系统界面发生的接收和处理消息和和程序调用的时间。

RT TTCN-3 引入了一个机制,在系统适配级别存储消息到达或者程序调用的时间。消息接收的时间点被系统适配自动注册,传达给测试可执行程序并存储该消息。时间消息可以通过 -> timestamp 来获得。

The existing redirections for getcall, getreply, receive, trigger, catch, and check operations are extended by an optional clause timestamp. A redirect specification of the form:

-> timestamp VariableRef

4. TTCN-3 性能和实时测试例子工程

```
testcase TC_wait() runs on Comp system Comp {
  /*
   * HINT: In RT TTCN-3 time progress is measured with a test system clock. The clock is
   * initialized (set to 0.0) at the beginning of each test case execution and is available during
   * the complete test run in each component. The clock values are represented as float
   * values.
   */
  var float refTs;
  var float ts;

  map(mtc:realtimePort, system:realtimePort);

  // log the current time
  log("now is ", now);

  refTs := 0.5;
  // wait for test system clock 0.5sec (e.g. the time since test case started in seconds)
  wait(refTs);
  // if wait before send, suspends the sending until specified wait time is reached
  realtimePort.send(3);

  log("message sent at ", refTs);

  wait(0.6);

  // store the timestamp when the message was received into ts
  realtimePort.receive(3) -> timestamp ts;
```



```
log("message received at ", ts);

log("Message timestamp difference (receive - send): ", ts - refTs);
if (ts - refTs > 1.0) {
    setverdict(fail, "Time limit exceeded: ", ts - refTs, " exceeded ", 1.0, " seconds");
} else {
    setverdict(pass, "Time constraint fulfilled: ", ts - refTs, " seconds");
}
unmap(mtc:realtimePort, system:realtimePort);
}

testcase TC_timestamp() runs on Comp system Comp {
    /*
    * HINT: In RT TTCN-3 time progress is measured with a test system clock. The clock is
    * initialized (set to 0.0) at the beginning of each test case execution and is available during
    * the complete test run in each component. The clock values are represented as float
    * values.
    */
    var float refTs;
    var float ts;

    map(mtc:realtimePort, system:realtimePort);

    // log the current time
    log("before send: ", now);

    realtimePort.send(3)
    // store the timestamp when the message has been sent into refTs
    -> timestamp refTs;

    log("message sent at ", refTs);

    wait(2.0);

    log("after wait: ", now);

    // store the timestamp when the message was received into ts
    realtimePort.receive(3) -> timestamp ts;

    log("after receive: ", now);
```

```
log("message received at ", ts);

log("Message timestamp difference (receive - send): ", ts - refTs);
if (ts - refTs > 1.0) {
    setverdict(fail, "Time limit exceeded: ", ts - refTs, " exceeded ", 1.0, " seconds");
} else {
    setverdict(pass, "Time constraint fulfilled: ", ts - refTs, " seconds");
}
unmap(mtc:realtimePort, system:realtimePort);
}
}
```