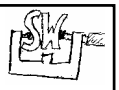
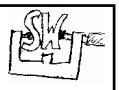


*TIMED*TTCN-3 – A Real-Time Extension for TTCN-3

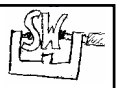
Jens Grabowski

Institute for Informatics,
Software Engineering Group
University of Göttingen



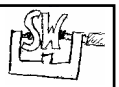
Outline

- Introduction
- The *TIMED*TTCN-3 Approach
- Example
- Summary, Status & Outlook



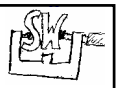
*TIMED*TTCN-3=TTCN-3+*Time Extensions*

- Non-functional black-box testing:
  - Testing of real-time requirements, e.g.:
    - Response time/latency, jitter, throughput, …
- TTCN-3 timer inadequate for real-time:
  - Intended to prevent blocking of test case
  - Snapshot semantics not designed for real-time



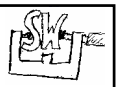
Description of Real-Time Requirements

- Separately from functional requirements
- Mathematical formulae, which describe the relationship of timestamps $t_{i.}$, e.g.:
  - Latency: $\forall i : t_{bi} - t_{ai} \leq 10ms$
- Implemented as TTCN-3 functions



The *TIMED*TTCN-3 Approach

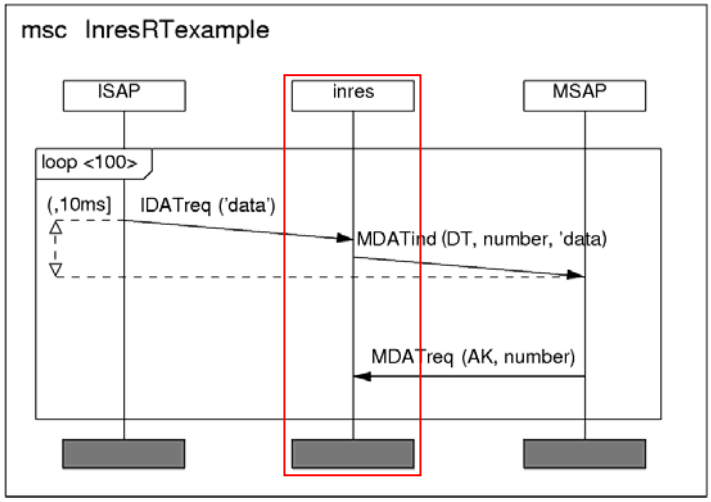
1. Instrument functional testcases to generate timestamps
2. Execute testcase
3. Apply evaluation functions to the generated timestamps:
   - Offline evaluation
   - Online evaluation
4. Assign a test verdict



TTCN-3 Extensions

1. **Local clock:** read, wait
2. **Logfile:** log, sort, retrieve
3. **Timezones:** specification of clock synchronised test components
4. **Conf-Verdict:** **`pass`⇨`conf`⇨`inconc`⇨`fail`**

Example



Testcase

User defined timestamp type

```
type record TimestampType{float logtime,
                          MessagesType messagename}

testcase InresRTexample
{
  var float sendTime:=self.now+5.0;
  for (i:=1; i<=100; i:=i+1)
  {
    resume(sendTime);
    log(TimestampType:{self.now, IDATreq});
    ISAP.send(IDATreqType:{"data"});
    MSAP.receive(MDATindType:{DT,expected_num,"data"});
    log(TimestampType:{self.now, MDATind});
    MSAP.send(MDATreqType:{AK,expected_num});
    sendTime:=sendTime+0.1;
  }
}
```

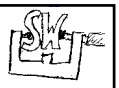
Read local clock

Wait until time point

Dump timestamp to log

Dump timestamp to log

Calculate next resume time point



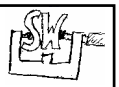
Logfile with Timestamp Objects

➔ **{5.619, IDATreq}** `logfile.first(TimestampType:{?,-},`
`TimestampType:{?,IDATreq})`

➔ **{5.627, MDATind}** `logfile.next(TimestampType:{?,MDATind})`

➔ **{5.632, IDATreq}** `logfile.next(TimestampType:{?,IDATreq})`

➔ **{5.641, MDATind}** `logfile.next(TimestampType:{?,MDATind})`



Offline Evaluation Function

```
function latency(logfile timelog, MessageType msg1,
   MessageType msg2, int count, float upperbound)
   return verdicttype
{
  var TimestampType stampA, stampB;
  var int i;
  timelog.first(TimestampType:{?,-},
                TimestampType:{?,msg1});
  for(i:=1; i<=count; i:=i+1)
  {
    stampA:=timelog.retrieve;
    timelog.next(TimestampType:{?,msg2});
    stampB:=timelog.retrieve;
    if (not (stampB.logtime-stampA.logtime<=upperbound))
    { return conf; }
    timelog.next(TimestampType:{?,msg1});
  }
  return pass;
}
```

Signals to match
Number of messages
Logfile parameter
Upper latency bound
Sort logfile, reset cursor
Retrieve timestamp
Advance cursor
Retrieve timestamp
Set verdict
Compare timestamps

Linking Execution and Evaluation

```
control
{
  var testrun myTestrunHandle, verdicttype myVerdict;

  myTestrunHandle:=execute(InresRTexample());

  myVerdict:=latency(myTestrunHandle.getlog,
                     IDATreq,MDATind,100,0.01);

  myTestrun.setverdict(myVerdict);
}
```

Handle for accessing logfile and verdict of testrun

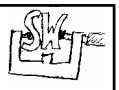
Execute instrumented testcase

Evaluate timestamps

Set testrun verdict according to evaluation

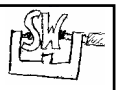
Summary and Status

- *TIMED*TTCN-3, a real-time extension for TTCN-3:
  - Local clocks, Logfile, Conf-verdict, (Timezones)
- Status:
  - *TIMED*TTCN-3 (Syntax, BNF, *TIMED*TTCN-3-GFT) has been submitted to ETSI for standardization.
  - Ideas of *TIMED*TTCN-3 found its way into the UML testing profile, but not into TTCN-3.
  - Methodology for using *TIMED*TTCN-3 in combination with real-time communication has been presented to TestCom 2004. This work has been submitted to the ETSI work item on test patterns.



Outlook

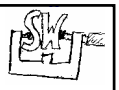
- Outlook:
  - Extend existing TTCN-3 operational semantics towards real-time
    (will only be done if *TIMED*TTCN-3 will find its way into standardization).
  - Refinement of methodology.
  - Extensions towards performance testing.



Credits

The presented work has been carried out together with:

Zhen Ru Dai
(Fokus Fraunhofer in Berlin)
and
Helmut Neukirchen
(University of Göttingen).



Thank you for your attention!

Any questions?