Property verification on traces

{emmanuel.gaudin,eric.brunel}@pragmadev.com
Project consortium

http://www.presto-embedded.eu/
Project objectives

• Improvement of industrial Real time Embedded Systems development process

• Early verification of functional and non-functional properties

• V1 – verify that the system model (M) satisfies the properties (P).

• V2 – verify that the system model (M) contains the traces (T) that have been monitored by observing the running system.

• V3 – verify that the execution traces (T) conform to identified properties (P).
Candidates to express properties

- Temporal logic
  - Ambiguous
  - Complex
- MSC – Message Sequence Chart ITU-T standard
  - Simple
  - Poor to express properties
- Property Sequence Chart
  - Unambiguous
  - Simple (similar to MSC and UML Interactions)
- UML MARTE
  - Activities
  - Interactions (similar to MSC)
Concept

- Use the same type of diagram for:
  - Traces
  - Specification
  - Properties
- In order to easily verify the properties on the traces
Traces

Message Sequence Chart
MSC
ITU-T Z.120
international standard

Execution traces:
• Instances,
• States,
• Messages,
• Operation calls,
• Semaphores,
• Timers,
• Actions.

Display of system time
Specification

A specification is an expected behavior for the system.

- Sequence of events
- Alternatives and loops
- Time constraints
A way to express causality:
• If *cause* then *effect*
• If *not cause* then *effect*
• If *cause* then *not effect*
• If *not cause* then *not effect*

With simple annotations on the MSC.
Property Sequence Chart

Component Instance c1

Fail message
f:a

Required message
r:a

Regular message
e:a

Parallel operator
par

Alternative operator
alt

Unwanted Message Constraint
b
b={C_i,l_1.C_j,...,C_k.l_n.C_t}

Wanted Chain Constraint
\Rightarrow g

Unwanted Chain Constraint
\Rightarrow g

Loop operator
loop(x,y)

Time-line

Strict operator

A picture is worth a thousand words
Tool: PragmaDev Tracer

- Free
- Windows, Solaris, Linux

PragmaDev Tracer is a free tool for graphical requirements, properties, and traces using a standard graphical representation. Main characteristics are:

- Express the requirements and properties of your system.
- Trace your system behavior on-line or off-line.
- Verify the trace matches the requirements and the properties.

**Technical characteristics**

- **Easy connection**
  Clocks to the tracer is done through a socket with a possibility of acknowledgement of each information received.
- **Integration in your testing or development environment**
  The tracer can be started in test mode and in batch mode and control commands can be received through the socket.
- **Integration with PragmaDev Studio**
  Applications generated by PragmaDev Studio code generator embed tracing information allowing to directly trace execution on target live through an IP connection.
- **Easy documentation**
  Generates PDF, PostScript and HTML or copy/paste to word processors for easy documentation.
- **Property match**
  PragmaDev Tracer can verify requirements and properties on the traces.
- **Flexibility**
  PragmaDev MSC Tracer runs on Windows and Linux.

**Features**

PragmaDev Tracer is a graphical tracing tool based on ITU-T Message Sequence Chart and on OMG UML2.0 Sequence Diagram standards. It provides graphical representations of:

- Tasks, States, Messages, Semaphores, Timers, Objects, Alternatives, Loops, Time constraints, Chain constraints.

Each event is related to system time information.

**Property definition**

PragmaDev tracer supports Property Sequence Chart (PSC) in order to define expected properties of a system. Within the PSC language, a property is seen as a relation on a set of exchanged system messages, with zero or more constraints. PSC may be used to describe both positive scenarios (i.e., the "desired" ones) and negative scenarios (i.e., the "unwanted" ones) for specifying interactions among the components of a system. PSC has both formal notation and operational semantics.

**On-line demonstration**

- See a 13 minutes online presentation and demonstration of the PragmaDev Tracer V2.0 here.

**Free download**

- Current stable V2.0 developed within the PRESTO European project as a.zip file (98 006 634 bytes) including all platforms (Windows, Linux, Solaris):
  PragmaTracerV2.0.zip
- Previous version V1.2 as a.zip file (49 715 256 bytes) including all platforms (Windows, Linux, Solaris):
  MiscTracerV1.2.zip
Main window

- Traces can be read from socket
- Traces can be read from files
- A new diagram can be created from scratch
- Existing diagram can be opened: RTDS XML, MSC-PR, OTF.
- Defines the socket port number
Editor / Tracer

- Preview
- Expand / Collapse
- Filters
- Copy / Paste
How to generate traces

• Model simulation (requires an executable model such as an SDL model)
• C macros for legacy code
  • Easy to insert in existing code
  • Easy to generate from a modeling tool
  • Write in a file or in a socket
  • File or socket can be read by the tracer
• MSC-PR, OTF, TTCN-3
Three levels of verification

- Basic diff for the same type of diagrams
- Specification against trace
- Property match based on PSC
A picture is worth a thousand words

Examples

UCAAT 2014
A picture is worth a thousand words
Use cases

• Thales Communications & Security - France
  ➢ Software Defined Radio (SDR) system
  ➢ TDMA MAC layer

• Thales - Italy
  ➢ Mobile Ad Hoc Network (MANET) in support of the Ultra-Wide Band (UWB) positioning system
  ➢ Selected component: OLSR (Optimized Link State Routing) level-3 network protocol
<table>
<thead>
<tr>
<th>Services</th>
<th>TDMA</th>
<th>NC</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDMA_Init</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMA_InitResult (Success)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMA_Status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time Request</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time Reply</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMA_Trafﬁc_Cfg (Tx)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start Timer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_RxMessage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_TxNpRequest</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMA_Trafﬁc_Cfg (idle)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TimerExpired</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMA_StatusResponse</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMA_StatusResponse</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TXData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AckTXData</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TDMA_Trafﬁc_Cfg (idle)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A picture is worth a thousand words