Eurofighter & A400M Programmes, Automatic Test Equipments

TESTBRICKS

Software framework for IEEE Standard 1641 test programs.

FERNANDO MUÑOZ MANRIQUE – Engineering Manager
SLGAT Meeting 2012-2 / MBDA Stevenage, UK / November 27, 2012
Presentation of the case study:

Developing a Complete Test Program Using IEEE 1641
Main conclusions (2007)

- The overall conclusion is that developing a test program based on the IEEE 1641 elements and philosophy is a valid, working and interesting approach, and a good alternative to traditional methods based on test actions.

- Probably the learning curve for an electrical/test engineer is less arduous than using other techniques (ATLAS, general-purpose languages).

- The portability of the products obtained is granted by the use of an IEEE standard.

- The most outstanding problem is the lack of a true programming language and/or environment to create complete test programs including BSC’s/TSF’s as native elements.
“Development with general purpose languages is **more complex and less productive** than using specific languages for test and measurement.”
For truly portable test requirements, **carrier language** ideally should be:

- Platform independent
- Signals natively supported
- Supported by development tools

Development process should be easy and straightforward
If ( adjust > 0 )

ampSignal (amplitude = 12v )

EndCondition

ampSignal.Out.Run(0)
WRITING THE TEST PROGRAM

BUILDING BLOCKS

Variables

Functions

Add new blocks
CREATING A SEQUENCE

1. Place blocks in a stack
   - var = var * exp
   - If(var < 10)

2. Configure parameters
   - var = var * exp
   - If(var < 10)

3. Read pseudocode from top to bottom
   - Real var = 2.5
   - var = var * exp
   - If(var < 10)
WRITING THE TEST PROGRAM

EDITOR
The virtual machine execution environment:

- **Editor**
- **Test Modules**
  - **XML**
  - **Interpreter**
  - **STD RTS**
    - **COTS Test Executive**
    - **ATE #1**
    - **ATE #2**
    - **ATE #3**

The Interpreter provides a virtual machine.
EXECUTION ENVIRONMENT

INTERPRETER

COTS Test Executive

<commands>
<status>

Test Executive Interface

Execution Manager

<shared resources>
<thread synchronization>

Variable Sharing Engine

STD Run Time System

<signals management>
Signals, allocators and drivers are dynamically loaded
DEBUGGER

FINDING ERRORS

Testbricks: Software framework for IEEE Standard 1641 test programs
## ATML SUPPORT

<table>
<thead>
<tr>
<th>ATML Components</th>
<th>TESTBRICKS FRAMEWORK UPGRADES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Description</td>
<td>Editor function to convert Test Description to pseudocode and to generate Test Description from pseudocode</td>
</tr>
<tr>
<td>Test Configuration</td>
<td>RTS Resource Manager to load dynamic assemblies depending on Test Configuration file</td>
</tr>
<tr>
<td>Test Station</td>
<td>Allocator assemblies to allocate signals to drivers based on data from Test Station, Instrument Description, Test Adapter and UUT Description files</td>
</tr>
<tr>
<td>Instrument Description</td>
<td></td>
</tr>
<tr>
<td>Test Adapter</td>
<td></td>
</tr>
<tr>
<td>UUT Description</td>
<td></td>
</tr>
<tr>
<td>Test Results Diagnostics</td>
<td>Responsibility of the COTS Test Executive (Not part of the framework)</td>
</tr>
</tbody>
</table>
CONCLUSIONS

- Create **STD compliant** test programs following an **easy and straightforward** process
- Benefit from less errors and more quality, more productivity and less cost
- Write test modules independent of the ATS in **pseudocode**
- Minimize the cost of instruments obsolescence and TPS rehosting with the interpreter **virtual machine**
Fernando Muñoz Manrique
Eurofighter & A400M Programmes, Automatic Test Equipments

escalade@indra.es / fmmanrique@ieee.org
@fmmanrique
http://www.linkedin.com/in/fernandomunozmanrique/en

C\ Mar Egeo, 4 Pol. Ind. N\º1
28830 San Fernando de Henares,
Madrid España
T +34 91 627 31 21
M +34 686 521 981
www.indra.es