Phase II Demonstration Overviews

ATML
• In last year’s demo we showed the feasibility of an end-to-end ATML-based software solution for TPS development and execution

• This year’s demonstration showed that the ATML standards are ready for use, and are being supported by increasingly mature software tools
  — The team worked with tool vendors to improve and integrate their ATML-compliant features

• The Demonstration focused on four key goals of the ATS Framework achievable through ATML
  — Supporting TPS lifecycle
  — Supporting TPS rehost
  — Supporting net-centric services
  — Tools support
The following companies participated in this demonstration:

- Agilent
- Boeing
- EADS
- Fox Software Limited
- Geotest
- Indra
- Lockheed Martin
- MAC Panel
- National Instruments
- Northrop Grumman
- PIDESO
- Rohde & Schwarz
- Summit Test Solutions
- Teradyne
- TYX
- Vektrex
- Virginia Panel Corporation
ATML Tools Catalogue

- Agilent/Vektrex (Booth 1300)
  - Instrument Description
  - Test Description Editor
- Boeing (Booth 601)
  - Use of Test Description, Instrument Description, Test Configuration, Test Adapter, and Test Station
  - Test Results Diagnostic Net Centric Services
- EADS (Booth 801)
  - newWaveX 1641 Signals, ATML Capabilities, Test description behaviour, Instrument Description Test Station description
  - PAWS test diagram tool
  - TRD System – Test Requirements Import/Export
- Geotest (Booth 811)
  - Converting ATML Demo files into ATEasy
- Lockheed Martin (Booth 1101)
  - TestConfiguration editor.
  - TestStation editor.
  - Test Description editor.
- National instruments (Booth 1001)
  - Test Description Importer (Test Stand, CVI)
  - Test Results Generator
  - PXI ATML Demo
- Rohde & Schwarz (Booth 806)
  - Modular Instrument Descriptions
  - ATML Capabilities
- Virginia Panel Corporation (Booth 711)
  - Instrument Description Analysis
ATML Demo Platforms

- DoD - ARGCS ATS2
- National Instruments - PXI System
- Geotest – BGATS
- Lockheed Martin – LM Star
- Boeing - RTCASS
The demonstration focused on the following key areas:

1. ATML Test Description supporting TPS life cycle
2. Modular instrument description within an ATS description
3. Interfacing with Test Results archiving databases
4. ATML Capability for a UUT with digital and bus testing
5. Test Diagram Generation
6. Use of Test Configuration at test station runtime and tool support for the generation of instance files
Key Areas

1. Test Diagram Generation
2. ATML Test Description supporting TPS life cycle
3. Modular instrument description within an ATS description
4. Use of Test Configuration at test station Runtime
5. Interfacing with Test Results archiving databases

- Test Diagram Generation
- ATML Test Description supporting TPS life cycle
- Modular instrument description within an ATS description
- Use of Test Configuration at test station Runtime
- Interfacing with Test Results archiving databases
ATML Demonstration
ARGCS ATS2 and NI PXI System
ATML Demonstration

Test Requirements Development
  • Lockheed martin TPS Builder
  • TYX TRD System
  • Vektrex Test Description Editor

Test Program Development
  • National Instruments TestStand & LabWindows/CVI
ATML Demonstration

1.1 Define Test Requirements using TYX TRD System
1.2 Define Test Requirements using Lockheed Martin TPS Builder
1.3 Define Test Requirements using Vektrex Test Description Editor
2. Import Test Description into National Instruments TestStand
ATML Demonstration

3. Implement code modules in LabWindows /CVI
   - Analog, Digital and Serial Bus operations
   - Using in principal IVI Driver calls
4.1 Execute Test Program – no fault example

Test Sequence Passed
4.2. Execute Test Program – fault injection example

![Repair Instructions]
Replace Transistor Q1 (Short)
Replace Transistor Q1 (Open)
Replace Resistor R1 (Open)

![UUT Result]
Test Sequence Failed
5.1. Generate test results in ATML format
5.2. Upload test results in central database
5.3. Post test results on Twitter (proof of concept for web upload)
Test Description supporting TPS life cycle
Supporting TPS Life Cycle

TPS

Test Requirements

ATML

Test Interface Adapter

Test Documentation

Rehost

Development

Maintenance

Test Program
1. Test Requirements Specification
2. Test Requirements Documentation
3. Test Requirements Traceability
Enable customers (DoD and Industry) to **formally specify** and **automatically enforce** the information contents of Test Requirements Documents for various milestones in the TPS development processes (SRR, PDR, CDR).
Operation “Other” allows simple descriptive text entry for PDR
Test Requirements Specification & Verification...

“Other” text is replaced with detailed operations for CDR.
Create human readable Test Requirements Documentation from ATML Test Description instance documents
Maintain consistency between test requirements and test program implementation, as test program changes during TPS integration and maintenance
Test Requirements Traceability...
Test Requirements Traceability...

<table>
<thead>
<tr>
<th>Stimuli:</th>
<th>Other:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Measurement Data:</th>
<th>Test Point</th>
<th>Signal Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measured Value</td>
<td>13.2 kOhm</td>
<td></td>
</tr>
<tr>
<td>High Limit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low Limit</td>
<td></td>
<td>12.2 kOhm</td>
</tr>
<tr>
<td>Supplemental Data</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Result</th>
<th>Go to Test</th>
<th>Adjust</th>
<th>Replace</th>
</tr>
</thead>
<tbody>
<tr>
<td>In Tolerance</td>
<td>T1100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Out High</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Out Low</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Modular Instrument Description within an ATS Description
• Uses IEEE 1641 Signal Models
ATML Capability Description

• Signal-based requirements [optionally expanded]
  Compared and matched to

• Signal-based capabilities
• Signal attributes are similarly transferred to the capability

\[
\text{amplitude} = \text{"-5 dBm"}
\]

• How can these values be transferred to the parameters of the resource?

  – Since, each controllable parameter of a resource maybe bound to more than one signal aspect

  – E.g. Three Phase Synchro

    Three sinusoidal phase attributes, each bound to a single attribute of rotor angle
• Resources may be described in terms of TSFs

• However:
  – TSF attribute formulae propagate towards the signal
  – Resource descriptions require signal properties to propagate towards the controllable resource attributes

∴ Capability descriptions require attributes
Modular Instrument Description

Matching Test Requirements to ATML Capability
Creating Instrument Calls

Compare:
<1641:Signal Requirement> == <atml:Capability> ?
Generate:
IVI Driver Code

Driver Code

IEEE 1641 Interface

Resource Manager

IEEE 1641 TSF Library

Capability Driver Description Library
• **1641 Demo II.V** and **ATML Demo** have trialled this process:

  - **Required 1641 Signal** is matched to an **ATML Capability of a given resource**
  - **Identified resource** is located in **Resource Driver Description**
  - **Each 1641 interface action** yields a set of **driver (IVI) commands or assignments**
  - **Process** cannot determine **whether the assignment value (e.g. ‘-20’), is a literal or a variable. <> denote a possible variable.**
  - **Doesn’t manage loops or conditional statements**
  - **Output is generated in a syntactically independent form, leaving it open to incorporation into a choice of programming languages**

---

**E.g. IVI Driver Commands & Assignments**

```
WXResource

RsRFSigGen.new
RsSpecAn.new

RsSpecAn.Initialize("FSG", true, true, ")
IRsSpecAnBasic = RsSpecAn.Personality.Selected("Basic")
RsRFSigGen.Initialize("SMJ", true, true, ")
IRsSpecAnBasic.Frequency.Spans = 0
RsSpecAn.System.IO.WriteScpi("CALC:MATH:FUNC:SUM:RMS:ON")
RsSpecAnBasic.Acquisition.DetectorType = RsSpecAnBasicDetectorTypeEnum.Rs
RsSpecAnBasic.Sweep:Coupling.SweepTime = 0.2
RsSpecAnBasic.Acquisition.SweepModeContinuous = false

RsRFSigGen.RF.Level = <20>
RsRFSigGen.RF.Frequency = <15000000000>
IRsSpecAnBasic.Frequency.Center = <15000000000>

<nominal> = <20 dB>
RsRFSigGen.RF.OutputEnabled = true
RsRFSigGen.RF.WaitUntilSettled(<3000>)
IRsSpecAnBasic.Level.ReferenceLevel = (RsRFSigGen.RF.Level + <nominal> + 2)
IRsSpecAnBasic.Trace:Initiate
IRsSpecAn.System.WaitForOperationComplete(<3000>)

<measurement_complete> = true
RsRFSigGen.RF.OutputEnabled = false
<measurement> = RsSpecAn.System.IO.QueryScpi20, "CALC:MATH:FUNC:SUM:"
Interfacing Test Results
Interfacing Test Results
Interfacing Test Results
Test Diagram Generation
ATML Demo - Test Diagram Generation

• Task to generate Test Diagrams automatically using ATML data

• Primarily focuses on TSFP and AFP Framework Key Elements
  • Test Station Functional and Parametric information (TSFP)
  ---- Description of test station capability
  • Adapter Functional and Parametric information (AFP)
  ---- Description of interface hardware used in a TPS

• Utilized ATML Test Station, Test Adapter, UUT Description, Test Description and WireLists schemas in the modified tools
  • Modification of tools to support the demo system, work remains for comprehensive ATML implementation

• TOWL test diagram tool (NAVAIR JAX)
  • After modifications uses ATML files to generate static test diagrams

• PAWS test diagram tool (TYX)
  • Modified to use ATML data and ATLAS to generate test diagrams at runtime using PAWS runtime system
ATML Demo - Test Diagram Generation

- ATML Tool
  - Lockheed
- Manual Process
- ATML Tool
  - Vektrex
- Manual Process
- ATML Wirelists
- ATML Test Station
- ATML Test Adapter
- ATML UUT Description
- ATML Test Description
- Generate Test Diagram
  - TOWL Tool, JAX
  - PAWS Tool, TYX
- Test Diagrams
  - (Runtime)
ATML Demo - Test Configuration at Test Station Runtime

• Task to use Test Configuration at runtime and tool support

• This Demonstration primarily focuses on the following Framework Key Element
  • Master Conformance Index (MCI)
    — Configuration information required to test, evaluate and maintain a UUT

• Utilized ATML Test Configuration file to verify instruments and interface hardware are present
  • Test Configuration parser to extract instruments and interface hardware
  • CVI User Interface module displays configuration verification results
  • TestStand sequence in Process Model to invoke verification

• Modified MTPSI tool to generate MTPSI file from ATML Test Configuration file (NAVAIR JAX)

• New ATML Test Configuration editor tool (Lockheed)
## Test Configuration User Interface

**Test Configuration Information**

### Hardware
- ITA
- UUT
- Interface cable

### Instruments
- Digitizer / Scope
- Power Supply
- DMM
- DTI
- Power Switch
- Medium Freq Switch
- Matrix Switch
- Function Generator

### Logical Names | Version | Availability
--- | --- | ---

**Test Program**

[OK button]
• **What have these demonstrations accomplished?**

  • Provided an application of the ATML standards in real world scenarios
  • Encouraged the development of new ATML tools
  • Demonstrated how tools that are targeted to proprietary or platform specific formats can be modified to use the open system ATML concepts
  • A use case with multiple ATML standards working in concert for one application
  • Further proving of the ATML standards during the trial use period
  • Provided benefits of ATML implementations to encourage adoption by industry
ATML Demo – Standards & Next Steps

• Test Diagram Demo – Used ATML Standards 1671.6 Test Station, 1671.5 Test Adapter, 1671.3 UUT Description
  • Standards were proven to work together to support data related to routing of signals
  • Demonstration identified a void which new WireLists schema filled

• Test Diagram tools were modified to specifically target the Demo system
  • A comprehensive ATML implementation requires further work
    — Test Diagram SBIR to provide a complete solution and include signal information
    — Demonstrating these tools on actual fielded TPS is a worthy goal

• Test Configuration Demo – Used ATML Standard 1671.4 Test Configuration
  • Standard proven to support the MTPSI use case and runtime asset verification applications

• MTPSI tool was modified to support conversion from ATML to MTPSI format
  • Additional work to modify MTPSI tool to convert MTPSI to ATML Test Configuration format
Lessons Learned
Lessons Learned

• **Transition to ATML Full-Use**
  - Added Wirelist.xsd to support Test Diagram Generation use case
  - Proposal to add Capability attributes (Capabilities.xsd)
  - Created extensive example for Test Description (IEEE 1671.1-2009)
  - Test Description recommendations during trial use period
  - Clarifying use of ATML Extension mechanism
  - Transition all ATML standards to common baseline schemas

• **ATML does help facilitate the use of COTS tools**
  - Significant reuse of ATML information and that used by the COTS tools used in the demonstration
    • TPS Support
    • Test Program Generation
    • Resource Selection
    • Diagnostics and Test results Archiving
    • Test Diagram Generation
    • Test Configuration checks
  - Users able to use standards with minimal outside assistance

• **ATML is ready for full use**
Lessons Learned

- There is a need to have a basic level of ‘standards adherence’ on ATS platforms.
- Adherence to standards need to be verified. e.g.
  - IVI Conformance
  - ATML File Validation (format and possibly data contents)
- Validate that systems contain a ‘minimum’ amount of information conforming to the ATS Framework Standards.
- Provide series of methods and processes that can verify correct use of standards
- As not all interfaces can be standardized, ATS acquisition should also specify, evaluate, and verify ATS software architectures which are critical for TPS development and long-term maintainability.
Recommendations

• **Recommendation #1**
  – The ATS Framework Working Group should identify potential DoD users and programs that are suitable as process owners and would benefit from ATML information exchange. The ATS Framework Group should encourage the ATE industry to demonstrate the feasibility of using ATML-enabled tools.

• **Recommendation #2**
  – The ATS Framework Working Group should continue to address key goals (e.g. supporting TPS life cycle and TPS rehost) using ATML targeted to projects identified by interested users.

• **Recommendation #3**
  – Provide expertise into new programs using ATML standards. The project team membership should include DoD stakeholders, various IPT representation e.g. NxTest & TPS IPTs, ATS Framework members, and key commercial organizations who have shown a commitment to the success of ATML.