Abstract—The IEEE Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML (IEEE Std 1671™-2010) [1] and all its ‘dot’ standards have been published and are available from the IEEE.

The ATML standards working group is now revising these trial use 'dot' standards to match their current XML schemas in line with the full use IEEE Std 1671.

The "Common" ATML XML schemas are posted to an IEEE download site on the World Wide Web, available for download and use. In short, the ATML standard and it’s ‘dot’ companions are now published, available, and their associated XML Schemas are downloadable from the Web, and in use across industry.

This paper provides:

a) An explanation as to the rational behind each of the ATML components; representing ATML companion ‘dot’ standards, as well as the use and reliance of existing IEEE standards IEEE Std 1641™ (STD) [2].

b) An explanation of how each companion ‘dot’ standard relates to each other, and how they may come together to provide the synergistic ATML Framework defined by IEEE Std 1671™-2010[1].

c) An explanation as to how each component ‘dot’ standard can be used standalone; and how each can be used with other component ‘dot’ standards to provide for ever increasing interoperable test systems.

d) An explanation on how ATML allows user extensions as a core mechanism to promote common Automatic Test System (ATS) information interchange

Finally, the paper explains the common misconceptions associated with test information interchange and explains how the ATML family of standards is an ideal solution for test information interchange across different and varied platforms, for test systems of today and into the future.

Keywords-ATML, Test Description, Instrument Description, UUT Description, Test Configuration, Test Adaptor Description, Test Station Description, Test Results, Signal Modeling, IEEE 1671, IEEE 1641, WireList, Capabilities.

I. INTRODUCTION

In 2010 the ATML 1671 Base standard was updated and revised into a IEEE Full Use five year standard. This base standard replaces the previous 2006 version and augments the ATML Family of ‘dot’ standards providing extra features and information.

The ATML family of standards now achieves both a level of maturity and, with the publication of the final ‘dot’ standard in 2009, the full width of test information support. The publication of the ATML 1671 Base standard represents a significant core descriptions of test information through its common schema definitions, which are now stable for the foreseeable future, and available for reuse by the existing ‘dot’ standard.

Using the ATML standards both individually and together represents the next phase of creating the full use standards to provide a complete set of normalized and integrated ‘dot’ standards utilizing these common schemas. Also within the IEEE SCC20 committee the stability of these common schemas is seen as a major contributory factor in helping all the SCC20 information standards (IEEE Std. 1232, IEEE Std. 1636, IEEE Std. 1671, IEEE Std. 1641, IEEE Std. 1505) come closer together in the future.

The key aspect to using ATML is not necessarily to consider new tools for building ATML files, but to figure out how to integrate ATML into existing tools and development environments using the ATML’s firm and stable baselines.

Critical to tool vendors is the ability to ensure that the ATML test information can be tailored to allow them to hold test information they use within the ATML format. As part of the ATML philosophy it provides an extensive set of extension mechanisms that once understood provide a convenient way to add new test information into the existing ATML formats, without compromising interchangeability of information, and through their use provide perfect vehicles for assess changes and revisions as the standard develops over the next 10-20 years.

II. ATML EXTENSABILITY

The ability to add additional ‘user defined’ Test information into the ATML exchange formats is seen as key design feature of ATML. There are three mechanisms that are provided within the standard for this purpose
• Wildcard-based extensions allow for the extension of the XML schemas with additional elements.

• Type derivation allows for extending the set of data types by deriving a new type from an existing common element type.

• Lists derived from c:NamedValues allowing user defined properties with attached values.

**Wildcard-based extensions** - the user defines their own XML schema and namespace and then uses the content within one of the many `<c:Extension>` elements within the existing ATML Standards.

**Type Derivation** - allows users to use their own XML schemas or to reuse existing ATML types from other domains ('dot' standards). Within these schemas are defined new types that are identified via xsi:type. These new types add the additional information to an existing type and represent good candidates for improving the standard when it revised.

Allowing extensibility with the ATML standard is a key design goal. Using extensibility however in any instance document, requires close securitization by the user. ATML does lay down rules of when extensions should not be used, and these need to be adhered to:

• Extensions shall not prevent application using the standard information

• Extensions shall not repackage existing information

• Extensions shall be associated with a user defined namespace

• The extension user defined schemas and documentation shall be provide with the instance documents that use the extensions

### III. HISTORICAL BACKGROUND

In 2002, the ATML focus group was formed (outside any formal standardization body) with a mission to ‘Define a collection of XML schemas that allows ATE and test information to be exchanged in a common format adhering to the XML standard’.

The scope of this effort was the standardization of test information to allow test program and test asset interoperability, as well as UUT test data (including results and diagnostics) to be interchanged between heterogeneous systems.

In 2004, the efforts of the ATML focus group were brought into IEEE Standards Coordinating Committee 20 (SCC20), where the formal standardization process has taken place. Further refinements and updates to the work accomplished by the ATML TII sub-committee has (and continues to) take place within both the ATML focus group and IEEE SCC20 subcommittees.

In 2009 the final ATML ‘dot’ standard was published to give us the complete set of ATML Family of standards published under the IEEE standards banner.

This year 2010 we’ve been focused on the normalization and revision process updating our Trial-Use standards into Full-Use five year standards. The first of these is the ATML base standards, which has been updated to include all the lessons learnt and common schemas identified as part of completing each if the ‘dot’ standards. The revised standard has given us a core set of common XML schemas that can be retrospectively used on the existing ‘dot’ standards and represent a mature and stable set of test information exchange formats.

### IV. ATML FOCUS

The goals for ATML are to:

• Establish an industry standard for test information exchange

• Develop a exchange format that can be understood by man or machine

• Allow, and design for, user extensibility

• Establish a process for managing extensibility

• Ensure acceptance within the user community

The ATML focus group is an accumulation of domain experts, who have come together with a common purpose to provide a practical solution. These experts utilized use cases and XML tools, to develop, and to further refine, ATML frameworks and their components.

The general uses cases were:

• Dynamic test sequences that can change with historical data

• Support instrument setup directly

• Support instrument setup using signal descriptions

• Support parallel/simultaneous testing and complex timing relationships

• Capture test results

• Capture test description information and sequencing

• Capture instrument specifications and capabilities

• Capture test station specifications and capabilities

• Capture test setup and test configurations

• Capture UUT specifications and requirements

• Capture test support hardware and software

• Capture UUT diagnostic Built In Test (BIT) codes

### V. ATML USAGE

The ATML initiative came about by a desire to standardize on the XML format, rather than the various proprietary tools used within the test industry. By using a common format, different tools and systems can exchange information and be brought together to form co-operative heterogeneous systems, which, through the use of ATML standardization, can:
• Decrease test times
• Reduce incidents of Can Not Duplicate or No Fault Found
• Reduce the repair cycle
• Formalize the capture of historic data which has been the preserver of experts in the field to heuristically identify faulty components
• Allows greater use of commercial products and tools

VI. ATML FRAMEWORK

One of the key aspects of the ATML initiative was to provide a framework to allow the information needed to support ‘Test’ to come together under one entity. The dilemma was that a lot of this information was already in the process of being or planned to be standardized, so rather than considering a single standard for all things. The concept of a family of related standards was born. Under the IEEE Std 1671 ATML Framework this would draw together the existing test standards, provide the necessary glue to allow consistent interchange and augment the existing standards with a new set of component (‘dot’) standards to address the remaining key ‘test’ elements

Within ATML, the term framework is used to differentiate the ATML effort from providing any specific architecture or software. The aim was for ATML to support a multitude of different architectures and allow interchange between different software, vendors or test systems.

ATML frameworks have been developed to:
• Summarize and organize the essential elements of an ATS.
• Provide a common frame of reference.
• Eliminate the need to use a variety of custom file formats.
• Provide compliance with the World Wide Web Consortium (W3C) standards to support Net-centric solutions.
• Be based upon open standards.
• Be extensible.
• Enable the creation of modular ATS architectures (components based upon the ATML component standards can easily be substituted and data can be shared between the components).

ATML frameworks are defined in the form of three distinct approaches:
• External interfaces
• Internal models
• Services requirements

A. External Interfaces

External Interfaces represent the information that exchanges between distinct subsystems. For ATML these subsystems are described within the ATML family of standards (See Table I)

The components of the ATML framework represent the key aspects within a typical ATS system. In many cases within one component, information from other components is often required, in which case one schema will make a reference to a components described by another ATML component schema.

The relationship between these components of ATML and a ‘generic’ ATE is depicted in the diagram in Fig 1.

B. Internal Models

Internal Models ensure a consistent approach to defining elements that need common semantics, across different entities. Within ATML there are several such models.
• Signal Definitions using 1641
• ATML Capabilities
• Pins Ports and Connectors
• wirelist and netlists

The use of these items across ATML components ensure that different elements interpret the same information in the same way, and describe similar components in a common format. Examples would be
• A signal defined for a test or as a capability of an instrument can be identified as the same.
• Wirelists for connecting hardware components use the same models.
• Defining Connectors used in Test Description, UUTs, ITA Test Station Cables

○ Services Requirements

For a test system solution using an ATML framework, there is a need to add specific services that use the ATML test information. It is recognized with ATML that the definition of the External Interfaces and Internal models is generally not enough to enforce heterogeneous cooperative test systems. A simple scenario of “Tell me your configuration” or “What is the next test” require that we not only define the information exchange but standardize on how the questions should be asked.

For ATML these services shall be defined by the implementer using the Services Description Language (WSDL), which provides for XML message based communication.
VII. ATML STANDARDS

The following table identifies the standards collectively known as the ATML family. The ATML family of standards (See Fig 2) can be used collectively to support ATML applications. The applications may use other standards to support addition frameworks such as the DoD’s ATS framework.

TABLE I. ATML STANDARDS

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
<th>Publication Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE Std 1671-2010</td>
<td>IEEE Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML</td>
<td>2010</td>
</tr>
</tbody>
</table>

Figure 1. ATML UML Model
A. IEEE Std 1671-2010 Common, HardwareCommon and TestEquipment, Capabilities, Wirelists and TestResults

ATML’s ‘Common’ provides the definition of common types and attributes that are used by more than one of the ATML XML schemas. Common, as a result, is simply a “toolbox” for other XML schemas.

IEEE Std 1671-2010 defines five common schemas: Common.xsd, HardwareCommon.xsd, TestEquipment.xsd, Capabilities.xsd and Wirelist.xsd.

Common.xsd provides ATML unique types and attributes, HardwareCommon.xsd provides ATML hardware unique types and attributes, TestEquipment.xsd provides test equipment unique types. Capabilities.xsd provides a method to define instrument and resource capabilities that can be referenced by the ATML ‘dot’ standards. WireList.xsd provides the interconnection information between hardware components described by the ATML ‘dot’ standards.

B. IEEE Std 1671.1 (InPress) Test Description

Test Description facilitates the definition of the test performance, test conditions, diagnostic requirements, and support equipment to locate faults, align, and verify proper operation of a UUT.

Signal descriptions within a Test Description can be represented utilizing IEEE Std 1641 (STD).

Test Description(s) are utilized in the development of a TPS.

C. IEEE Std 1671.2-2008 Instrument Description

Instrument Description facilitates the definition of the static description of an Instrument. The Instrument Description will facilitate the descriptions of a single instrument, or of synthetic, virtual and composite instrumentation, both describing the instrument physical characteristics and specification but also its capability within the test system.
The IEEE Std 1671.2-2008 standard also defines an instrument instance schema to augment the static instrument description, and to allow actual instrument instance information serial numbers, etc. to be exchanged.

D. IEEE Std 1671.3-2007 UUT Description

UUT Description facilitates the unique description of a particular UUT. This includes information such as the name, part number, model number, type, power requirements, interfaces, physical properties, and operational requirements.

E. IEEE Std 1671.4-2007 Test Configuration

Test Configuration facilitates the identification of all the hardware, software and documentation necessary to test a UUT on a particular ATS.

F. IEEE Std. 1671.5-2008 Test Adapter

Test Adapter facilitates the unique description of the interface between the UUT and the Test Station, the physical and electrical characteristics, the capabilities / performance, the identification and classification, etc. This includes the cables, connectors, wires, contacts, etc.

G. IEEE Std 1671.6-2008 Test Station

Test Station facilitates the specification of a particular automatic test station. This includes the physical and electrical characteristics such as; the paths between test system ports and the instruments; tolerances and accuracy of the test station; test station identification information such as part number, serial number, nomenclature, and location; status information such as calibration data, dates, and self test status; operational history, such as system up-time; external interfaces; safety information such as interlocks and temperature sensing; power requirements; controller definitions; etc.

H. IEEE Std 1636.1-2007 Test Results (and Session Information)

Test Results and Session Information provides the definition for the data collected that resulted from executing test(s) of a UUT in manual, semi-automatic or fully automatic test environments. This includes the measured values, pass/fail results, and accompanying data including test operator, station information, environmental conditions, etc.

I. IEEE Std 1641™-2010 Signal and Test Definition

All signal components used with ATML can be referenced through the IEEE Std 1641 Signal & Test Definition standard. Signals are used in ATML in the following areas: Test Description, Instrument Description, Test Station Description, and Test Adapter.

VIII. CONCLUSION

The ATML family of standards (as well as XML schemas) represents a major breakthrough in both government and industry cooperation, providing for a universally unique family of mature standards for the exchange of diverse and varied test information in a common manner. The potential for newly developed common tools and processes that can utilize one or more of the ATML standards is potentially unbounded, while providing some standard design methods to the more traditional and expected test information problems. The ATML family of standards and XML schemas marks a major advancement in both the cooperation of competitive companies as well as represents a model approach of how to facilitate genuine standard advancement and intercompany cooperation.

The new ATML Framework standard, its companion ‘dot’ standards (and their associated XML schemas) are all available. The popularity of ATML can already be seen by its adoption in commercial off the shelf (COTS) products, which is already fostering test information interchange across competing vendors.

Looking forward, as the ATML family of ‘dot’ standards are updated to full use, they will all be normalized to use the baseline mature ATML common schemas and provide a truly integrated and consistent approach for exchanging test information across COTS software tools as part of an Open System Architecture and ATML Framework solution.

REFERENCES