Applications of IEEE P1671.1 ATML Test Description

Ion A. Neag
Reston Software
12000 Market St. Ste. 204
Reston, VA 20190
(703) 234-2932
ion.neag@restonsoftware.com

Mike Seavey
Northrop Grumman Corporation
600 Hicks Road
Rolling Meadows Illinois 60008
(847) 259-9600 x4696
michael.seavey@ngc.com

Abstract – The emerging ATML Test Description standard (IEEE P1671.1) defines an XML-based format for describing automated test programs. This paper contains an overview of the ATML Test Description format, provides insight into its design, and suggests a number of applications that have the potential to improve development and maintenance processes for automated test programs.

I. INTRODUCTION

The IEEE Standards Coordinating Committee 20 on Test and Diagnosis for Electronic Systems (SCC20) is currently conducting the balloting process for the Automatic Test Markup Language (ATML) family of standards. ATML defines a standard exchange medium for sharing information between components of Automatic Test Systems (ATSs), including test data, resource data, diagnostic data, and historic data. The exchange medium is defined using the eXtensible Markup Language (XML) [1].

Within the ATML family, IEEE P1671.1 defines a draft standard exchange format for test description information, in support of the development of Test Program Sets (TPSs) used in an automatic test environment [2]. In the following, this format will be called "ATML Test Description"

II. THE ATML TEST DESCRIPTION FORMAT

A. Purpose

The information that can be encoded using the ATML Test Description format describes test performance, test conditions, diagnostic requirements, and support equipment necessary to locate, align, and verify the proper operation of a Unit Under Test (UUT). The use of a standard format for test descriptions will promote interoperability between components of Automatic Test Systems (ATSs) and will facilitate the preparation of TPS documentation [2].

The current technology landscape in ATS software for aerospace and defense is characterized by the prevalence of legacy TPSs written in ATLAS, which will have to be maintained for a long time, as well as the emergence of new test development technologies based on general-purpose programming languages and test executives. The ATML Test Description format is intended to support both approaches and the transition of legacy TPSs towards new technologies.

To understand the possible applications of the standard, it is important to note that the description of test behavior occurs at a higher level of abstraction, compared to the one typically required for describing an executable TPS. The ATML Test Description format is not intended to be directly executable, or directly compilable to an executable format. In other words, an ATML Test Description document[1] describes "what" needs to be tested, but not "how" to perform the test.

Some elements of the ATML Test Description format are specifically designed to support the signal-oriented testing paradigm, where test actions are described in terms of signals applied or measured at the pins of the UUT [3]. This paradigm is extensively used in aerospace and defense ATS applications and is considered by many as the best solution for long-term TPS maintainability. The existence of elements supporting signal-oriented testing does not preclude the use of the format for describing TPSs that use an instrument-oriented paradigm.

[1] In XML terminology, an "XML document" may be a physical file, XML data transmitted through a network connection, XML data stored in a database, etc.

B. Contents

An XML document that conforms to the ATML Test Description format can contain some or all of the following data items[2] [3]:

1) References to documents, drawings, diagrams, parts lists associated with the UUT

2) Connector information for the UUT

3) Performance characteristics of the UUT

4) Power and signal requirements for testing the UUT

5) Possible faults or failures of the UUT and the associated repair information

6) The list of Tests to be performed on the UUT

   6.1) Parameters that customize the behavior of each Test, including the values of scalar parameters and the description of stimulus signals

   6.2) Test Results produced by each Test, including the identification of the signal or scalar value to be measured and the limits of nominal operation

   6.3) Preconditions that must be fulfilled for each Test, before the Test can be executed

   6.4) Optionally, the behavior of each Test, described as a set of signal-oriented Actions such as Setup, Connect, Measure, etc. Alternatively, the behavior can be described through free-form text or in a custom XML format.

7) The sequence of Tests necessary to detect UUT faults

8) Optionally, the sequence of Tests necessary to isolate UUT faults (e.g., a "fault tree"). Alternatively, this sequence can be determined by a diagnostic reasoner in accordance with a diagnostic model defined for the UUT [4].

III. Design

A. Requirements and Use Cases

The XML schema for ATML Test Description was designed by the ATML Working Group, which includes participants from the Government and Industry representing various classes of users: software tool vendors, TPS developers, system integrators, system maintainers, etc. The design was provided to the IEEE SCC20, which is responsible for creating and maintaining the standard document [6] [7].

The design is based on an extensive set of requirements and use cases identified by the Working Group. An essential design requirement was to support the development of new TPSs, as well as the maintenance and rehosting of legacy TPSs. This requirement has led to a set of *very diverse use cases*, including:

1) Description of test programs written in the ATLAS language.

2) Description of test programs written in modern general-purpose programming languages, or created with graphical programming tools.

3) Description of test programs executed under the control of test executives.

4) Description of test requirements supported by legacy Test Requirements Documentation (TRD) formats.

To accommodate these diverse use cases, the XML schema offers a high degree of flexibility regarding the information to be included in particular XML documents. For example, one document may describe Tests as "black boxes", specifying only the Parameters and Test Results of each Test, while another document may describe the detailed behavior of each Test using Actions.

B. Relationship to Other ATML Component Standards

An important requirement for ATML Test Description was consistency with other component standards from the ATML family, specifically IEEE Std. 1636.1 (Test Results) and IEEE Std. 1671.3 (UUT Description) [8]. This is achieved through cross-referencing of XML documents, consistent naming and organization of various XML elements and attributes, and shared use of XML types from the Common schema [1].

The relationship between ATML Test Description and other ATML component standards is illustrated in Figure 1 and described in the following.

1) AN ATML Test Description document describes a test program, but is not involved directly in its execution (although exceptions may exist). Each execution of the test programs produces an ATML

[2] This is not an exhaustive list.

Test Results document. For closed loop diagnostics applications it is important that various elements of the ATML Test Results document can be traced back to the corresponding elements of the ATML Test Description document. For example, each test from the test program is described by a "Test" element in ATML Test Description. Each execution of the test produces a "Test" element in ATML Test Results. This "Test" element will contain an identifier that references the "Test" element in ATML Test Description, describing the executed test.

2) The ATML Test Description document describes the features of the UUT relevant for TPS development. Alternatively, it can reference a UUT Description document, which describes the UUT in more detail.

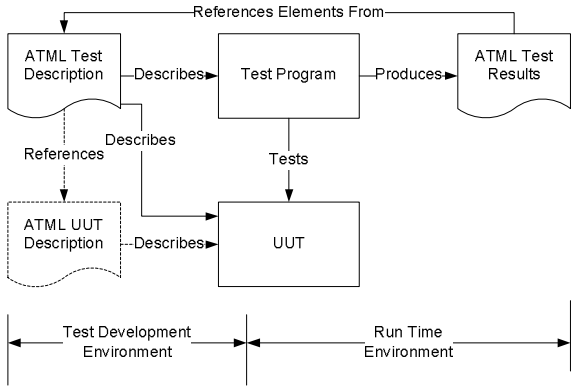

Figure 1 – Relationship between ATML Test Description and other ATML Component Standards

C. Relationship to Other ATS Standards

The ATML Test Description schema was required to be compatible with other current and retired ATS standards, as follows:

1) ATML Test Description can be used in conjunction with IEEE Std. 1232 AI-ESTATE [5] to support intelligent diagnostics applications, as described later in this paper. Consistency with the XML schemas that are currently being developed for IEEE Std. 1232 AI-ESTATE is ensured through consistent naming and structuring of XML elements and attributes.

2) ATML Test Description can be used to describe test programs compliant with IEEE 1641-2004 Standard for Signal and Test Definition [9]. The ATML Test Description schema uses XML types defined by IEEE Std. 1641 to describe stimulus signals and signal measurements. This feature is illustrated in Figure 2. In addition, the Action types defined by ATML Test Description are compatible with the signal statements of the IEEE Std. 1641 Test Procedure Language (TPL).

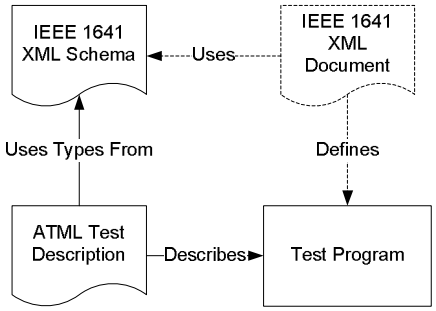

Figure 2 – Relationship between ATML Test Description and IEEE Std. 1461 XML

It is important to note that the IEEE Std. 1641 compliant signal definitions and Actions included in ATML Test Description are of general value and can be used to describe any signal-oriented test program, even if not compliant with IEEE Std. 1641.

3) The ATML Test Description format is intended to fulfill the requirements documentation role of retired TRD standards such as Mil-Std-1519 [10] and Mil-Std-1345B [11]. Many of the TRDs now in existence comply with one of the above standard formats, or to an extended format created for a specific program or organization. To ensure the usability of these legacy TRDs for future TPS re-hosting, the ATML Test Description format follows the general structure of the above TRD standards and supports all their data elements.

D. Extensibility

The ATML framework contains a powerful extensibility mechanism [1] [12], allowing users to combine in a single XML document standard-compliant information and non-standard information, while providing a well-defined mechanism to separate the two categories. The non-standard information can be specific to a particular application, tool or organization, or may be agreed upon between multiple organizations. The structure of non-standard information is typically defined in an XML schema supplied by the organization that creates the extension.

The design of the ATML Test Description schema attempts to accommodate the diversity of existing and future automatic test programs, while striving to reach a balance between capabilities and usability. Use cases considered infrequent or atypical are not supported

directly, as this would have added complexity to the schema and thus reduced its usability. However, the design of the schema enables users to add support for such use cases by providing "extension points".

For example, the standard format describes test behavior at a relatively high level of abstraction and thus ATML Test Description documents are not directly executable. Applications that require an executable XML-based format can be accommodated by embedding executable XML contents in standard-compliant XML documents, using the previously described extension mechanism. The resulting document can be executed by a specialized run-time engine that interprets both standard and extended XML contents. In addition, its standard XML contents can be used by any standards-compliant tools, for example to generate Test Requirements Documentation. The extension is made possible the extension point shown in Figure 3. Similar extension points are present in many other locations in the schema.

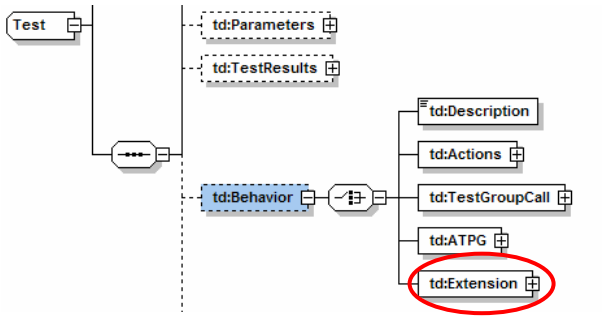

Figure 3 – Extension Point for User-Defined Description of Test Behavior

IV. APPLICATIONS

A. *Test Program Development*

In general, the development of test programs involves two different areas of expertise. On one hand, the UUT designers, familiar with the operation of the UUT, are in the best position to define *what* needs to be tested. Test engineers, familiar with the operation of the ATE platform and the software technologies involved, are required to define *how* to implement the test program. The transfer of information between designers and test engineers is often a significant bottleneck in the TPS development process.

The existence of a standard data transfer format, in conjunction with a well-defined TPS development process, can greatly improve the collaboration between UUT designers and test engineers. In such a scenario, illustrated in Figure 4, UUT designers use a data entry interface to create ATML Test Description documents in XML format. In the future, ATML Test Description documents may be generated automatically by design tools. Test engineers may convert the XML documents to traditional paper documents to be used as requirements specifications. Alternatively, they may process the XML documents with a code generator that creates a part of the test program (for example, the definitions of Tests and their sequencing) [1]. As in most cases the XML documents will not contain the level of detail necessary to generate complete test programs, the remaining code will be created manually by the test engineers.

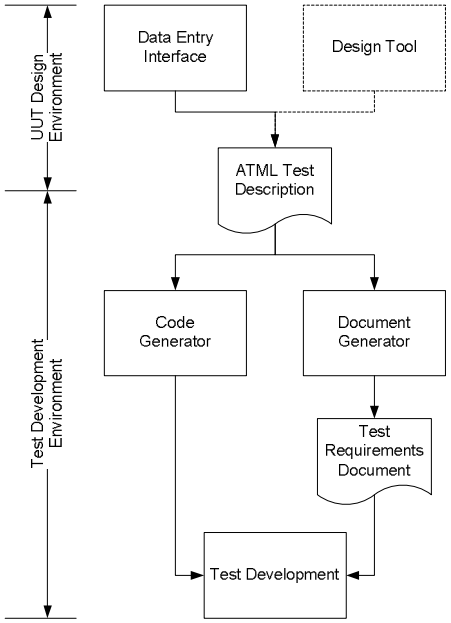

Figure 4 – Application of ATML Test Description for Test Program Development

B. *Test Program Documentation*

The rehosting of many legacy TPSs is difficult and expensive due to the lack of adequate TPS documentation. The existence of a standard TRD format, in conjunction with adequate contractual requirements, may alleviate this situation for newly developed TPSs. At this time there is no active standard for TRD formats. The ATML Test Description format intends to fulfill this need by supporting the test requirements information traditionally supplied through TRD documents.

For *new TPSs*, the ATML Test Description documents are an automatic byproduct of the test development process described in the previous section. ATML Test Description documents can be produced for *legacy TPSs* through engineering analysis, automated

extraction from TPS source code or, more likely, through a combination of these approaches [1]. To be usable for documentation purposes, the XML documents may be converted to common electronic document formats such as the Portable Document Format (PDF).

Being an electronic document format rather than a paper-oriented format, ATML Test Description provides greater flexibility in use. For example, it is no longer subject to the field length limitations that existed in traditional TRD formats. As the XML format does not specify the structure of the generated documents and their page layouts, it can easily accommodate the various internal document formats used by system integrators and maintenance organizations.

Traditional TRD formats provide a free-form text field for describing text behavior. This limitation can produce incomplete TRD documents and cause problems during TPS implementation and validation. Tool vendors and maintenance organizations tried to overcome the above limitation by defining formats to be used when describing test behavior in the free-form text area of TRD pages. While useful, these non-standard formats don't allow the exchange of data between tools and between organizations. The ATML Test Description format attempts to improve this situation by specifying a standard format for describing Test behavior through signal-oriented Actions, as illustrated in Figure 5.

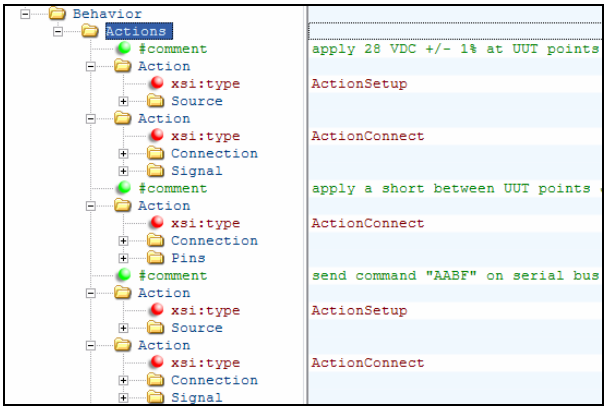

Figure 5 – Application of ATML Test Description for Specifying TPS Requirements

C. Test Program Rehosting

Most TPS rehosting efforts are based on the original test program implementation, complemented in some cases by TPS documentation in the form of TRDs, which are more or less consistent with the current version of the implementation. As current TPSs are typically implemented with tight coupling between components, and these components are specific to a particular architecture, *implementation-based re-hosting* is often an expensive and time-consuming process [2] [13].

The ATML Test Description format allows *requirements-based re-hosting*, where the requirements of the original TPS are encoded in a generic, implementation-independent form, then used to create a new implementation on the new ATE platform. As shown in Figure 6, the requirements description for the original TPS may have been created during its development, may be imported from existing TRDs, or may be extracted from the original implementation. Furthermore, the creation of the new implementation may be completely manual, or may be supported by code generation tools.

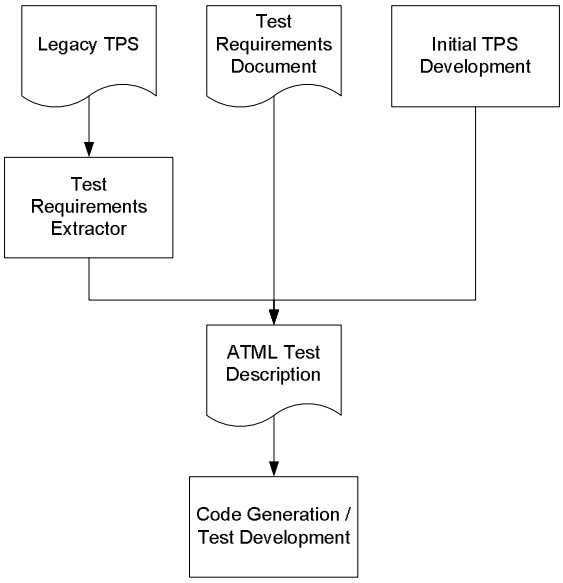

Figure 6 – Application of ATML Test Description for Requirements-Based TPS Rehosting

As it describes test behavior at a higher level of abstraction, the ATML Test Description format can be used to rehost TPSs on new ATE platforms that have a different run-time software, possibly supporting a different implementation paradigm. This feature is important for the long-term maintenance of TPSs in aerospace and defense applications, as it enables rehosted TPSs to make use of up-to-date software technologies.

D. ATE Specification

As ATE platforms become obsolete, maintenance organizations face the challenge of specifying

replacement platforms that support both the existing set of legacy TPSs and future TPSs to be developed for new systems. In this process, determining the performance requirements for a large set of legacy TPSs is a difficult endeavor. Requirements must be first extracted from individual TPSs, then aggregated at the ATE level. Given the number and complexity of legacy TPSs, these functions will likely require specialized software tools. Software tools will not extract 100% of the requirements from the legacy TPSs. The actual percentage will be based upon how "well" the legacy TPS was written and documented. TPS engineers will still be needed to fill in the remaining information into an ATML Test Description document.

The ATML Test Description format supports the ATE specification process by enabling the representation of requirements for individual signal operations, for individual Tests, and for the entire TPS. The later case is illustrated in Figure 7. The existence of a standard format will facilitate the interoperability of various software tools and will provide a standard way of delivering requirements information to end users.

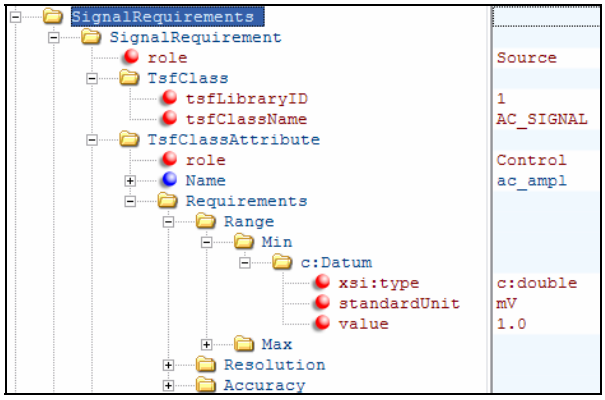

Figure 7 – Representation of Signal Requirements in ATML Test Description

E. Intelligent Diagnostics

Modern Information Technology and Enterprise Architectures enable maintenance organizations to collect and analyze historical test, diagnostic, and maintenance data. In this context, intelligent diagnostic reasoners are being considered as a solution for improving diagnostic accuracy and efficiency. IEEE Std. 1232 AI-ESTATE facilitates the interoperability of such reasoners by defining a standard exchange format and a standard set of services [5].

To enforce the separation of test and diagnostic functionality, the data exchange format defined by AI-ESTATE operates at a high level of abstraction, where Tests are considered "black box" entities. Consequently, the inputs, outputs and behavior of Tests are outside the scope of AI-ESTATE. However, this information is required to implement the test programs supporting the operation of intelligent reasoners. The ATML Test Description format complements the AI-ESTATE format by supporting the description of test information. Furthermore, the design of the ATML format enables users to combine reasoner-based diagnostic functionality with fixed, user-defined sequences. Such sequences may implement initial TPS functionality such as UUT identification, safe-to-turn-on tests, firmware upload, etc., or may refine the accuracy of diagnosis beyond the capabilities of the reasoner. A possible solution that integrates AI-ESTATE diagnostic models and ATML Test Description documents is shown in Figure 8.

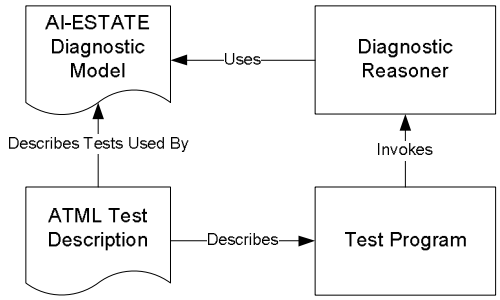

Figure 8 – Application of ATML Test Description and AI-ESTATE for Intelligent Diagnostics

F. Web Services

The integration of Service Oriented Architecture (SOA) elements such as Web Services in ATS applications facilitates the implementation of distributed systems in support of Network Centric Operations (NCO). In addition, SOA improves software interoperability through platform neutrality and loose coupling between components.

SOA Web Services are commonly specified using the Web Services Description Language (WSDL), which in turn is based on XML. Consequently, elements of various ATML formats can be used to define Web Services for ATS software architectures. For example, a Web Service based on the XML Schemas defined by ATML Test Description and ATML Test Results (IEEE 1636.1) may provide access to test runtime services, as exemplified in [1].

V. CONCLUSIONS

The emergence of the ATML Test Description standard has the potential to improve existing processes for TPS development and maintenance.

As a common format for documenting UUT test requirements, it can improve the interaction between design engineers and test engineers and enhance long-term TPS maintainability.

By facilitating software interoperability, the standard can create opportunities for new classes of applications such as computer assisted TPS rehosting and closed-loop diagnostics, where higher-level functions are obtained by combining the functionality of several interconnected software products.

As ATML Test Description documents are intended to be produced and consumed by software tools rather than humans, most of the standard's end users will be shielded from the complexity of the XML schema design. On the other hand, to achieve an effective utilization of the standard in applications users should become familiar with its main data modeling capabilities: classes of information supported and their relationships, constraints and limitations, relationship with other component standards from the ATML family, extension capabilities, etc. Several examples to be included in an Annex of the standard will support this need by providing detailed explanations and guidance.

VI. References

[1] IEEE Std. 1671™-2006, *Trial-Use Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML*

[2] IEEE, *Project Authorization Request for IEEE P1671.1, Trial-Use Standard Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML: Exchanging Test Descriptions*, downloaded from http://standards.ieee.org/board/nes/projects/1671-1.pdf , June 2006

[3] Neag, I.A., *Supporting Hardware Independence in the Next Generation of Automatic Test Equipment*, Proc. Autotestcon 2005, Orlando, FL

[4] ATML Working Group, *ATML Test Description schema*, downloaded from http://grouper.ieee.org/groups/scc20/ATML/TestDescription.xsd , June 2006

[5] IEEE Std. 1232™-2000, *Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE)*

[6] IEEE, *Home Page of the SCC20 Test and ATS Description (TAD) Subcommittee*, http://grouper.ieee.org/groups/scc20/td/index.html

[7] IEEE, *IEEE P1671.1™/D0.4 Draft Trial-Use Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML, Exchanging Test Descriptions*, working document of the SCC20 Test and ATS Description (TAD) Subcommittee

[8] IEEE, *Home Page of the SCC20 Test Information Integration (TII) Subcommittee*, http://grouper.ieee.org/groups/scc20/tii/

[9] IEEE Std. 1641™-2004, *Standard for Signal and Test Definition*

[10] U.S. Air Force, *Preparation of Test Requirements Document*, Mil-Std-1519, Washington, DC, 1 August 1977

[11] U.S. Navy, *Preparation of Test Requirements Document*, Mil-Std-1345B, Washington, DC, 10 February 1981

[12] Lopes, T.P., Neag, I.A., Ralph, J.E., *The Role of Extensibility in Software Standards for Automatic Test Systems*, Proc. Autotestcon 2005, Orlando, FL

[13] Neag, I.A., Hyde, J., Ramachandran, N., *Resolving ATLAS Obsolescence Problems - Lessons Learned and New Opportunities*, Proc. Autotestcon 2004, San Antonio, TX