Automatic Test Markup Language

<ATML/>

Sept 28, 2004

Contents

1 Introduction

1.1 Mission Statement

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.

1.2 Scope

ATML is the standardization of test information to allow test program and test asset interoperability, and UUT test data including results and diagnostics to be interchanged between heterogeneous systems.

1.3 Purpose

The Purpose of the document is to provide a single document to provide an overview of the ATML goals and aims and a flavor of the solution. In addition this document outlines the other documents that will be of use to the less casual reader.

This document provides guidance for the other ATML documents that make up the ATML standard.

1.4 Goals

- Establish an industry standard for test information exchange
- An 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

1.5 General Use Cases

- 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 program information and sequencing
- Capture instrument specifications and capabilities
- Capture test station specifications and capabilities
- Capture test setup and test configurations
- Capture Unit Under Test (UUT) specifications and requirements
- Capture test support hardware & software
- Capture UUT diagnostic and maintenance information

1.6 ATML usage

The ATML initiative has come about by a desire to standardize on the XML format used by 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.
- Close the loop on diagnostic systems

1.7 Convention for word usage

1.7.1 Shall, should, may and can

The word *shall* is used to indicate mandatory requirements strictly to be followed in order to conform to ATML and from which no deviation is permitted. The phrase *is required to* is equivalent to the word *shall*. The use of the word *must* is depreciated and shall not be used when stating mandatory requirements; *must* is used only to describe unavoidable situations. The use of the word *will* is depreciated and shall not be used when stating mandatory requirements; *will* is only used in statements of fact.

The word *should* is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain course of action is deprecated but not prohibited. The phrase *is recommended than* is equivalent to the word *should*.

The word *may* is used to indicate a course of action permissible within ATML. The phrase *is permitted* is equivalent to the word *may*.

The word *can* is used for statements of possibility and capability, whether material, physical, or casual. The phrase *is able to* is equivalent to he word *can*.

1.7.2 That and Which

The words *that* and *which* are commonly misused; they are not interchangeable. *That* is best reserved in essential (or restrictive) clauses, *which* is appropriate in nonessential (or non restrictive), parenthetical clauses. As a simple guide if a comma can be inserted before the word *that* or *which*, the word should be *which*. If a comma would not be used, the word to use is *that*.

2 System Framework

The ATML framework is defined in the form of two distinct approaches.

- External interfaces
- Web Services

Within this framework the External interfaces define domain orientated information that are used by Web Services. The information should always reference to the UUT or Repair process. The framework also defines, Web Services to generate, consume and manipulate this information.

The key framework point is that ATML is about information modeling, where the information models are expressed in XML

Extensibility is the driving focus for ATML initiative and is a key objective.

ATML is more than a minimum sub-set for today's knowledge, but the ATML Schemas concentrate on that common subset of information.

The ATML framework defines the components from which users can build their architectures, whilst being interoperable with other compliant architectures

2.1 External Interfaces

External Interfaces represent the information that exchanges between two distinct subsystems. For ATML these subsystems are defined as:

- TestResults
- Diagnostics
- TestDescription
- Instrument
- TestConfiguration
- UUTData
- TestStation
- InterfaceAdapter

This information *shall* be representable in XML and defined through the use of XML Schema Document (XSD) conforming to the XMLSchema specification

The relationship between the various components of ATML and a "generic" ATE is depicted in the following diagram.

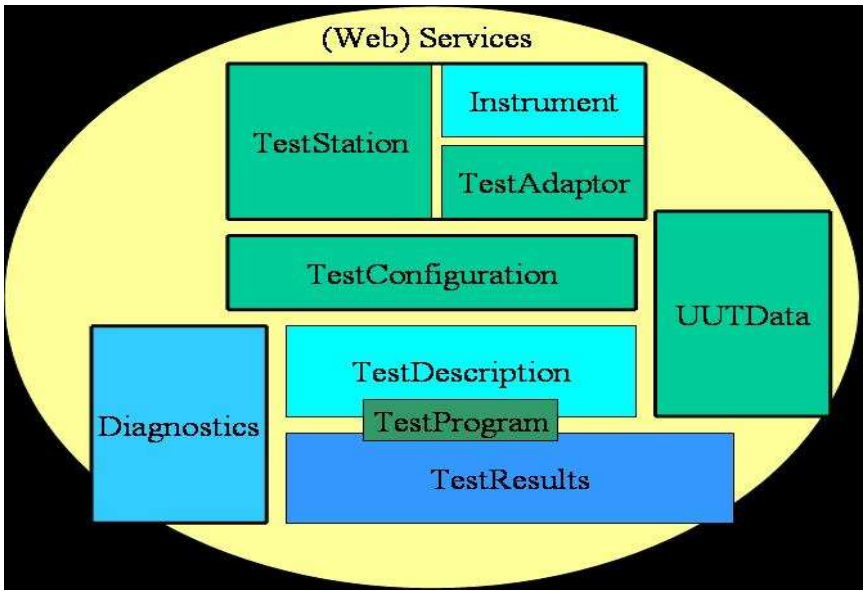

Figure 1 – ATML generic ATS Subsystems

2.2 Web Services

It is recognized within ATML that definition of the External Interfaces are generally not enough to enforce heterogeneous co-operative ATS systems. A simple scenario of "*Tell me your Test Configuration?"* or "*What is the next Diagnostic?*" require that we not only define the format of the information exchange but we can standardize on how the questions should be asked.

The Web Services infrastructure is founded on communication via XML-based messages that comply with a published Web Service description. The service description is an XML document written in an XML grammar called WSDL (Web Service Description Language) that defines the format of messages the Web Service understands. The service description serves as an agreement that defines the behavior of a Web Service and instructs potential clients in how to interact with it.

Where ATML components need to define such services, these shall be defined using WSDL.

2.2.1 **Web Service Description Language (WSDL)**

WSDL describes a Web service in two fundamental stages: one abstract and one concrete. Within each stage, the description uses a number of constructs to promote reusability of the description and separate independent design concerns.

At an abstract level, WSDL describes a Web service in terms of the messages it sends and receives; messages are described independent of a specific wire format using a type system, typically XML Schema.

An *operation* associates a message exchange pattern with one or more messages. A *message exchange pattern* identifies the sequence and cardinality of messages sent and/or received as well as who they are logically sent to and/or received from. An *interface* groups together operations without any commitment to transport or wire format.

At a concrete level, a *binding* specifies transport and wire format details for one or more interfaces. An *endpoint* associates a network address with a binding. And finally, a *service* groups together endpoints that implement a common interface.

3 Document List

3.1 Documents and XML Schemas release process

To introduce a new schema or document for incorporation into the ATML standard, a member *shall* submit the document to the ATML membership at a scheduled ATML meeting. In the case of a schema, the introducer must submit a preliminary use case and requirements document. A simple majority of the attending members plus the *approval* of the *chairman* is enough to introduce the document into ATML group as a working document (Draft).

Status:

1. Intro	– Initial introduction to ATML membership
2. Rejected	– Rejected after review by ATML membership
3. Draft	– Approved and under development within the working group; for schemas, schema development may proceed
4. Candidate	– Ready for testing (schema) or approval (document), minor revisions made following review
5.a Approved	– For documents NOT REQUIRED as part of the IEEE official documentation process. ATML members approved, ready for internal use by members
5.b Recommendation	– For documents (including Schemas) that will become part of the IEEE standard. ATML members approved, ready for publication through the ATML namespace, and submittal to IEEE for balloting
6. Release	– Balloted standard of IEEE

Draft is the initial working state of all ATML documents and schemas. For a Schema, the Intro and Rejected states *shall not* apply. Schemas *shall* begin this process at the Draft stage, and work on schema development shall occur only after use cases, requirements, transaction diagrams etc. have progressed to an *approved* state.

For use cases, requirements, transaction diagrams etc. these should be described using industry standards and best practices e.g. UML.

Candidate is intended for internal evaluation and testing by the ATML membership.

For a Schema to be a **candidate** there *shall* be a written specification that describes the semantics and components; every element, attribute, type etc. *must* be internally annotated. A **candidate** Schema *shall* also have at least two test cases identified for implementation that include all the major examples covering the use cases in the requirement specification. Each identified test case shall be owned by a working group member, who is responsible or validating the test cases. A **candidate** *shall* meet the

style guide requirements. A **candidate** may reference (include) other schemas that are at the draft level.

When a document becomes **candidate** the document *must* be posted to the ATML.org website. There is only ever one **candidate** version of a document (the one on the website), prior to becoming a **recommendation** with an official version identifier.

Recommendation *shall* have standard level documentation associated with the schema, and shall only reference other documents that are at the candidate level.

A **Recommendation** *shall* have all requirements met and *shall* have evidence that the test case implementations work. The owner of each test case shall provide evidence in the form of a report outlining how the test cases have been validated.

A major or minor change to a document is such that the major number changes when a new release of the document is made, a minor number changes at all other times. In general, minor changes are defined as typographic corrections or modifications to non-structural aspects of the schema (i.e. annotation changes). Major changes will be defined as structural modifications to the schema document.. Major version changes for Schemas shall be when changes to the schema would invalidate an instance document.

Version:

<major>.<minor>- Both <major> and <minor> are integer numbers; <minor> shall be two digits with a leading zero as required.

3.1.1 Approvals

Prior to ATML joining a standards organisation or if the organisation's procedures do not define an approval process for working documents, the following conditions *shall* be met when the term *approved* is used.

- To approve a **Draft** release – 75% of the working group membership plus a majority of currently assigned sub-schema working group chairpersons, with the ATML chairperson having the deciding vote

- To approve a **Candidate** release – 75% of the working group membership plus a majority of currently assigned working group chairpersons, with the ATML chairperson having the deciding vote

- To approve a **Recommendation** release – 75% of the general ATML membership plus a majority of currently assigned working group chairpersons, with the ATML chairperson having the deciding vote

- Otherwise the use of approved means - A majority of the attending members of the meeting plus the vote of the *chairperson.*

The working group membership at any meeting comprises of those attendees that have attended two out of the last three meetings (inclusive of the current meeting). All members that fulfil these requirements may vote. The chairperson of a working group may choose to seek all votes for the purpose of approval. This rule allows voting by proxy.

3.2 ATML Documents

3.2.1 Requirements

The ATML top level requirements will be captured in a version controlled document called "ATML Requirements and Use Cases". This will list detailed Use Cases and user requirements. In addition each ATML component will have a separate Requirements and Use Cases, see ATML Element Schemas Document Set.

3.2.2 Glossary

The ATML Glossary will be an extensive document detailing every major ATML tag and attribute name. The purpose of this document is to provide a common data dictionary and lookup for names and attributes used within the ATML system.

3.2.3 Style Guide

The Style Guide relates the style of ATML schemas; this is as compared to any style of the other ATML documents, which are described in this document. The Style Guide provides a common set of rules and references to other XML style documents that allows all ATML schemas to follow a common style and look and feel.

The Style guide will identify the preferred use of attributes and tags and defines rules such as:

The version on the ATML Schema shall be recorded using the <xs:schema version> attribute.

3.3 ATML Component Schemas

Each ATML component is supported by a set of documents based upon the ATML IEEE Document Template, which shall contain:

- An ATML Requirement and Use case specification section in the '*ATML Requirements and Use Cases*' covering each component plus cross-reference to Use cases.
- ATML Schema Specification containing support information, requirements, use cases etc. (.doc)
- XML Schema (.xsd) for defining information content

- WSDL Service definition for defining component Web Services

The process for submitting a draft schema or description document is that there shall be an associated **candidate** ATML requirement and use case specification for the component.

The main XML document is defined by the following table and represents the document types that all test information should use to be transmitted.

The ATML Component Schemas shall be a version controlled item and have a meaningful name with description and versions. The ATML components represent only items that could be instantiated in a document type or used as a sub element of another element. With the exception of the ATMLCommon schema a maximum of one root element definition per ***.xsd*** file. Each of the ATML sections shall represent a single entity with a top level node describes as follows:

1. **Common**
2. **TestResults** –contains a single run of test results
3. **Diagnostics** – describes a signal diagnostic entity
4. **TestDescription** – describes a test program
5. **Instrument** – describes one instrument
6. **TestConfiguration** – describes the configuration for testing
7. **UUTData** – describes a UUT
8. **TestStation** – describes one Test Station configuration
9. **InterfaceAdaptor**– describes one test adaptor

3.3.1 Common

Scope

Throughout ATML, there are types (complex and simple) and attribute groups that are shared among schemas. The Common sub-schema will contain these types. Additions to Common shall be made when components are identified as being used in multiple schemas.

Purpose

Any item that is used or may be used in more than one ATML sub-schema will be defined in the ATML Common schema. ATML Common shall contain only type or attribute group definitions and shall not include any elements definitions. There may be components included in the Common schema but not used in multiple ATML documents. Such components are included based on a determination that they may be shared across ATML sub-schemas. Common components are equivalent to common data types. Inclusion of Common in another sub-schema shall use a defined namespace to clearly identify different extensions.

3.3.2 TestResults

Scope
The TestResults schema is the source schema for all instance documents that will contain test results for a unit under test or for calibration of an ATE. TestResult instance documents will be the required format for data transfer between systems, and may be used as a data storage format, as desired by the user.

Purpose
The Test Results schema provides a standard format for publishing test results. This will permit such data to be easily shared for a variety of purposes, including statistical analysis and diagnostics. To provide a single source for the essential information about a test, the TestResults schema provides data elements to capture identifying information for: the UUT, the test station, the test program (via reference to a Test Description document); ambient environmental conditions at the time of the test, test equipment calibration data, test program input data (i.e. parameters) and ancillary textual comments.

3.3.3 Diagnostics

The proliferation of open distributed architectures and advanced diagnostic and analytical methods has given rise to a need for standard means to exchange diagnostic information between software components and applications. The Automatic Test Markup Language (ATML) Working Group has undertaken the task of developing a standard which fulfills this need.

Scope
The scope of this specification is to define in XML a set of interfaces for the interchange of information to support the execution and analysis of *diagnosis* and *diagnostic procedures*.

Purpose
The purpose of the ATML Diagnostic Schemata specification is to facilitate the sharing of diagnostic information between geographically and temporally distributed software applications that cooperate to accomplish the execution or analysis of diagnostic procedures.

This specification will provide XML Schemata and supporting information that allow the exchange of diagnostic information between conforming software components applications. The overall goal is to support loosely coupled open architectures that permit the use of advanced diagnostic reasoning and analytical applications.

3.3.4 TestDescription

Scope
The scope of this specification is to define in XML a set of interfaces for the interchange of information defining the test performance, test conditions, diagnostic requirements, and support equipment to locate, align, and verify proper operation of an Unit Under

Test (UUT). These test descriptions will be used in the preparation or documentation of *test program*(s). These test descriptions consist of *tests, test sequences*, *outcomes,* and *limits*; where:

A **Test** contains a set of stimuli, either applied or known, combined with a set of observed responses and criteria for comparing these responses to a known standard.

A **Test Group** is an unordered collection of **Test**s. The listed order of the **Test**s within a **Test Group** implies nothing about the sequence of execution.

A **Test Program** is defined as an implementation of the tests, test methods, and test sequences to be performed on a UUT to verify conformance with its test specification with or without fault diagnosis and is designed for execution on a specific test system. (Reference IEEE 100 definition 3 of "Test Program")

A **Test Sequence** is a specific order of related **Test**s.

A **Test Outcome** is a mapping from an observation to one of a set of discrete possibilities.

A **Limit** is the extreme of the designated range through which the measured value of a characteristic may vary and still be considered acceptable.

Purpose
The purpose of the ATML Test Description Schema specification is to facilitate the sharing of information necessary to:

- Test the performance of the UUT in accordance with the characteristics described by detailed performance characteristics, as well as to detect and indicate all faults and out-of-tolerance conditions.
- Adjust and align the UUT (when applicable).
- Isolate all detectable faults.
- Isolate each fault to an individual component or to the smallest group of components.
- Allow tests to be as simple as possible, independent from each other, and logically arranged to simplify testing and to eliminate redundancies.

This specification will provide an XML Schema and supporting information that allows the exchange of the test description information between conforming software components applications. The overall goal is to support loosely coupled open

architectures that permit the use of advanced test program generation, test program execution, and diagnostic reasoning applications.

3.3.5 Instrument

Scope

The Instrument schema is the source schema for all instance documents that contain a static description of an instrument model. A static description of an instrument contains data applicable to an instrument model rather than a single instance of a specific instrument.

Purpose

The purpose of the Instrument schema is to facilitate the sharing of information necessary to:

- Describe a tester configuration
- Verify the tester configuration at runtime to determine if the required instruments are present
- Allow for runtime or design time resource and switch path allocation
- Instrument selection
- Describe a synthetic (virtual/composite) instrument

Each instance document contains a static description of a single instrument model. The static description includes the topology (I/O and switching) and the capabilities of the instrument. The Instrument schema provides a structure for describing instrument capabilities. The Instrument schema does not define the elements and attributes used to describe the capabilities; it merely provides an extensibility point for referencing external schemas in which the elements and attributes are defined.

3.3.6 TestConfiguration

Scope

The TestConfiguration schema will encompass all information necessary to identify all of the hardware, software and documentation that may be necessary in order to test and diagnose a Unit Under Test (UUT) on a test system. The information may include, but is not limited to, test system assets used, hardware to electrically adapt or physically support the UUT on the test system, test program description software, test program media, maintenance manuals, and theories of operation.

Purpose

The TestConfiguration schema provides a framework which enables static test program set data to be exchanged between compliant systems. The data supports the aquisition and allocation of test assets required to be in place prior to testing a UUT on the test system.

3.3.7 UUTData

Scope
Provides a means of describing the UUT. This includes the static and dynamic UUT data including part number, serial number, nomenclature, operational requirements including hardware I/O and operations history.

Purpose
Instance documents derived from the UUTData schema shall provide complete documentation for the subject UUT. The UUTData schema will provide a means of capturing UUT historical information in addition to identification data. An instance document for a given UUT is intended to be the definitive source of information about that UUT.

Each UUT instance document will provide a unique identifier for its subject UUT This identifier shall provide a referential link to other ATML documents related to the subject UUT.

3.3.8 TestStation

Scope
TestStation documents shall contain descriptions of the paths between test system ports and the Instruments. Other information includes test station identification information such as part number, serial number, nomenclature, location; status information such as calibration data, dates, and self test status; and operational history, such as system up-time.

Purpose
The purpose of a TestStation instance document is to provide system documentation, including system drivers, calibration requirements, and all items that are related to any test station configured in the same way. All data that are related to a specific instance or a specific test station will be contained in TestStation.

This document will also contain information about the tolerances and accuracy of the test equipment and the Instrument revision.

The TestStation will also hold instrument instance data and status information such as calibration data, dates, self test status, at the time of the document being generated.

3.3.9 InterfaceAdapter

Scope
Describes the paths between test system ports and UUT pins.

Purpose
InterfaceAdapter documents shall contain descriptions of the paths between test system ports and the UUT. These descriptions will include the following definitions:

Cables, connectors, wire, and contacts, end to end configurations.
Mass Interconnect: Receiver, ITA, Cable Assemblies, Pre-Wired Adapters, Modules, Contacts, Connectors to instruments and the path through the Mass Interconnect to the UUT, and Enclosures for the Mass Interconnect to UUT connection.

All of the descriptions will include information on the manufacturer, the model and revision numbers, size and maximum capabilities of the device (this will include the number of contacts for connectors and the number of modules for the mass interconnects.) The InterfaceAdapter schema will also include interconnection cables. The definition for a cable will include the connector(s) used, the wire or cable used for the connection, a connection list defining the connections being made, and the contact(s) information.

4 ATML Standard and Conformance

The ATML Standard shall consist of the XML Schemas and their associated Schema Specifications for the Common components and each of the ATML Components.

An ATML instance document is considered conformant if the XML document is validated by the ATML schemas and adheres to all requirements specified in the relevant Schema Specifications. An ATML Document may be conformant with an individual ATML Component.