IEEE STANDARDS ASSOCIATION IEEE

IEEE Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML

IEEE Standards Coordinating Committee 20

Sponsored by the
IEEE Standards Coordinating Committee 20 on
Test and Diagnosis for Electronic Systems

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

20 January 2011

IEEE Std 1671™-2010
(Revision of
IEEE Std 1671-2006)

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1671-2010

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

IEEE Std 1671™-2010
(Revision of
IEEE Std 1671-2006)

IEEE Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML

Sponsor

IEEE Standards Coordinating Committee 20 on Test and Diagnosis for Electronic Systems

Approved 30 September 2010

IEEE-SA Standards Board

Approved 19 July 2011

American National Standards Institute

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1671-2010

Abstract: This document specifies a framework for the automatic test markup language (ATML) family of standards. ATML allows automatic test system (ATS) and test information to be exchanged in a common format adhering to the extensible markup language (XML) standard.
Keywords: ATE description, ATE test results, ATML, ATS, automatic test equipment, automatic test markup language, automatic test system, interface test adapter, ITA, SI, synthetic instrumentation, test configuration, unit under test, UUT description, UUT maintenance, XML instance document, XML schema

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2011 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 20 January 2011. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Incorporated.

IVI is a registered trademark in the U.S. Patent & Trademark Office, owned by the IVI Foundation, Inc.

W3C is a registered trademark of the World Wide Web Consortium (W3C) (registered in numerous countries). Marks of W3C are registered and held by its host institutions: Massachusetts Institute of Technology (MIT), European Research Consortium for Information and Mathematics (ERCIM), and Keio University, Japan.

PDF: ISBN 978-0-7381-6456-4 STD97016
Print: ISBN 978-0-7381-6457-1 STDPD97016

IEEE prohibits discrimination, harassment and bullying. For more information, visit http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html.
No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1671-2010

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation, or every ten years for stabilization. When a document is more than five years old and has not been reaffirmed, or more than ten years old and has not been stabilized, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon his or her independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Recommendations to change the status of a stabilized standard should include a rationale as to why a revision or withdrawal is required. Comments and recommendations on standards, and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Authorization to photocopy portions of any individual standard for internal or personal use is granted by The Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1671-2010

Introduction

This introduction is not part of IEEE Std 1671-2010, IEEE Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML.

Historical background

In 2002, an automatic test markup language (ATML) focus group was formed (outside any formal standardization body) with a mission to "define a collection of XML [extensible markup language] schemas that allows ATE [automatic test equipment] 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, which would allow for test program (TP) and test asset interoperability, as well as unit under test (UUT) data (including results and diagnostics), to be interchanged between heterogeneous ATE systems.

In 2004, the efforts of the 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 focus group has (and continues to) take place within both the ATML focus group and IEEE SCC20.

IEEE 1671 ATML family of standards

The ATML family of standards supports TP, test asset, and UUT interoperability within an automatic test environment.

This document provides an overview of the ATML goals, defines the ATML framework, defines the ATML family of standards, and specifies common ATML data elements, and common ATML schemas.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyright © 2011 IEEE. All rights reserved.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1671-2010

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Standards Association Web site at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA Web site at http://standards.ieee.org.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/reading/ieee/updates/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or nondiscriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Copyright © 2011 IEEE. All rights reserved.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1671-2010

Participants

The following Test Information Integration (TII) Subcommittee members contributed to the development and preparation of this standard:

Chris Gorringe, *Co-Chair*
Teresa Lopes, *Co-Chair*

Anthony Alwardt
Malcolm Brown
Matt Conrish
Timothy W. Davis
Robert Fox
Jose Gonzalez
Michelle Harris
Ashley Hulme
Anand Jain
Ion Neag
Dan Pleasant
William Ross
Mike Seavey
Dave Sharone
Ronald Taylor

The following member of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Stephen Allen
Anthony Alwardt
H. Stephen Berger
Malcom Brown
Keith Chow
David Droste
James Dumser
Sourav Dutta
Anthony Geneva
Chris Gorringe
Randall Groves
Werner Hoelzl
Ashley Hulme
Anand Jain
Adam Ley
Teresa Lopes
Robert Mcgarvey
Mukund Modi
Ion Neag
Leslie Orlidge
Gordon Robinson
David Rohacek
Bartien Sayogo
Mike Seavey
John Sheppard
Gil Shultz
Joseph Stanco
Walter Struppler
Ronald Taylor
Timothy Wilmering
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 30 September 2010, it had the following membership:

Robert M. Grow, *Chair*
Richard H. Hulett, *Vice Chair*
Steve M. Mills, *Past Chair*
Judith Gorman, *Secretary*

Karen Bartleson
Victor Berman
Ted Burse
Clint Chaplin
Andy Drozd
Alexander Gelman
Jim Hughes
Young Kyun Kim
Joseph L. Koepfinger*
John Kulick
David J. Law
Hung Ling
Oleg Logvinov
Ted Olsen
Ronald C. Petersen
Thomas Prevost
Jon Walter Rosdahl
Sam Sciacca
Mike Seavey
Curtis Siller
Don Wright

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish Aggarwal, *NRC Representative*
Richard DeBlasio, *DOE Representative*
Michael Janezic, *NIST Representative*

Lisa Perry
IEEE Standards Program Manager, Document Development

Soo H. Kim
IEEE Standards Program Manager, Technical Program Development

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

Contents

1. Overview 1
1.1 General 1
1.2 Scope 2
1.3 Purpose 2
1.4 Application 3
1.5 Conventions used in this document 4

2. Normative references 6

3. Definitions, acronyms, and abbreviations 7
3.1 Definitions 7
3.2 Acronyms and abbreviations 10

4. Automatic test system (ATS) architecture 12
4.1 Automatic test equipment (ATE) 12
4.2 Test program set (TPS) 15
4.3 Automatic diagnosis and testing 18

5. Automatic test markup language (ATML) 19
5.1 ATS architecture elements addressed by ATML 20

6. The ATML framework 22
6.1 External interfaces 22
6.2 Internal models 23
6.3 Services 23

7. ATML specification techniques 25
7.1 ATML common element partitioning 25
7.2 ATML XML schemas 28
7.3 XML schemas and their use in ATML 28
7.4 UML models 28

8. The ATML framework subdomains 29
8.1 The ATML framework and ATML family component standards 29
8.2 ATML subdomains 29

9. ATML XML schema names and locations 36

10. ATML XML schema extensibility 39

11. Conformance 40
11.1 ATML family XML schemas 40
11.2 The ATML framework 40

Annex A (normative) XML schema style guidelines 46
A.1 Naming conventions 46
A.2 XML declaration 48
A.3 ATML namespaces 48
A.4 Versioning 50
A.5 Documentation 51
A.6 Design 52

Copyright © 2011 IEEE. All rights reserved.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1671-2010

Annex B (normative) ATML common element schemas 55
B.1 Common element schema—Common.xsd 55
B.2 Common element schema—HardwareCommon.xsd 136
B.3 Common element schema—TestEquipment.xsd 227

Annex C (normative) ATML internal model schemas 258
C.1 ATML internal model schema—Capabilities.xsd 258
C.2 ATML internal model schema—WireLists.xsd 260

Annex D (normative) ATML runtime services 267
D.1 Messages 267
D.2 Executive system service 267
D.3 Example WSDL service definition 268

Annex E (informative) Pins, ports, connectors, and wire lists in ATML 269
E.1 Introduction 269
E.2 Overview of the base types 270
E.3 Using ports, pins, and connectors together 273
E.4 Ports, pins, and capabilities 275
E.5 Wire lists 278

Annex F (informative) ATML capabilities 283
F.1 Introduction 283
F.2 Overview 285
F.3 Describing instrument capabilities 289
F.4 Describing ATS capabilities 328
F.5 Capability information in ATML Test Description 332

Annex G (informative) IEEE download Web site material associated with this document 339

Annex H (informative) ATS architectures 340
H.1 ATS architectures utilization of published standards 340
H.2 ATS architectural relationships to IEEE SCC20-based standards 343
H.3 ATS architectural ATML subdomain relationship to SIMICA standards 343

Annex I (informative) Architecture examples 347
I.1 Instruments 347
I.2 Test descriptions 348
I.3 Complete testing scenario 350
I.4 Integrated ATML system 363

Annex J (informative) UML models 367
J.1 Generic ATS testing of a UUT 367
J.2 ATML XML schema relationships 369

Annex K (informative) Glossary 372

Annex L (informative) Bibliography 375

Copyright © 2011 IEEE. All rights reserved.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1671-2010

IEEE Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML

IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

1. Overview

1.1 General

Automatic test markup language (ATML) is a collection of IEEE standards and associated extensible markup language (XML) schemas that allows automatic test system (ATS) and test information to be exchanged in a common format adhering to the XML standard.[1]

The ATML framework and the ATML family of standards have been developed and are maintained under the guidance of the Test Information Integration (TII) Subcommittee of IEEE Standards Coordinating Committee 20 (SCC20) to serve as a comprehensive environment for integrating design data, test strategies, test requirements, test procedures, test results management, and test system implementations, while allowing test program (TP), test asset interoperability, and unit under test (UUT) data to be interchanged between heterogeneous systems.

[1] This information is given for the convenience of users of this standard and does not constitute an endorsement by the IEEE of this consortium standard. Equivalent standards or products may be used if they can be shown to lead to the same results.

Copyright © 2011 IEEE. All rights reserved.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1671-2010

1.1.1 ATML framework referenced IEEE standard

The ATML framework can reference IEEE Std 1641™ [B29].[2] This referenced IEEE standard, when utilized, is then considered part of the ATML framework.

1.1.2 Application of this document's annexes

This document includes twelve annexes. Of these twelve, four are normative (Annex A, Annex B, Annex C, and Annex D).

Annex A contains style guidelines for the ATML family XML schemas. Annex A guidelines shall be followed by ATML XML schema developers and maintainers during the development and maintenance of all ATML family XML schemas, including the XML schemas associated with this document.

Annex B contains XML schema element description and definitions for the **ATML common element XML schemas**. Annex B shall be utilized by ATML XML schema developers, maintainers, and ATML users. Annex B shall be referenced during the development and maintenance of all ATML family XML schemas, including the XML schemas associated with this document.

Annex C contains XML schema element description and definitions for the **ATML internal model XML schemas**. Annex C shall be utilized by ATML XML schema developers, maintainers, and ATML users. Annex C shall be referenced during the creation and development of ATML Capabilities or ATML WireLists documents.

Annex D contains guidelines for ATML services. Annex D shall be referenced by ATML users implementing an ATML framework.

Annex E through Annex L are informative and thus are provided strictly as information for both users and maintainers of this document.

1.2 Scope

ATML defines a standard exchange medium for sharing information between components of ATSs. This information includes test data, resource data, diagnostic data, and historic data. The exchange medium is defined using XML. This standard specifies the framework for the family of ATML standards.

1.3 Purpose

The purpose of ATML is to support TP, test asset, and UUT interoperability within an automatic test environment. ATML accomplishes this through a standard medium for exchanging UUT, test, and diagnostic information between components of the test system. The purpose of this standard is to provide an overview of ATML goals, define the ATML family of standards, and specify common data elements for the ATML family of standards.

[2] The numbers in brackets correspond to the numbers of the bibliography in Annex L.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

1.4 Application

1.4.1 General

This document should be applied anywhere ATS and test information is to be exchanged. This ATS and test information includes the following:

- Data that will be utilized for the design, development, and utilization of automatic test equipment (ATE).
- Data that will be utilized for the design, development, and utilization of test program sets (TPSs) to test a product (e.g., UUT) on a particular ATE.
- Product design data that will be utilized during the testing of the product (e.g., UUT).
- Shared usage of maintenance data and the results of testing a product (e.g., UUT).
- Testing requirements of a particular product (e.g., UUT).
- Data that will be utilized for the design, development, and utilization of instrumentation that will be utilized within a particular ATS configuration.
- A definition of allowable ATS configurations that can be use to test and evaluate a particular product (e.g., UUT).
- A definition of the capabilities of ATSs as well as the elements of the ATS.

1.4.2 Users

Anticipated users of the ATML family of standards include the following:

- Product (e.g., UUT) developers
- Product (e.g., UUT) maintainers
- TPS developers
- TPS maintainers
- ATE system developers
- ATE system maintainers
- Instrumentation developers
- Developers of ATML-based tools and systems
- Developers of prime mission equipment that use the supported UUT as a component

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

1.4.3 Precedence

In the event of conflict between this document and an ATML family component standard, this document shall take precedence.

In the event of conflict between this document and a normatively referenced standard (*Extensible Markup Language (XML) 1.0*),[3] the normatively referenced standard, as it applies to the information being produced, shall take precedence.

In the event of conflict between this document's XML schema definition and/or annotations and an ATML family component standard and/or XML schemas, this document's XML schema definition and/or annotations shall take precedence.

1.5 Conventions used in this document

1.5.1 General

Within the body of this document, the conventions defined in Table 1 are utilized.

Table 1—Document conventions

Item	Convention
The use of **bolded** text	Emphasizes a word or concept.
The use of *italics*	Represents bibliography references and quoted text from other documents.
The term "ATML framework"	Represents the sub-domains of an ATS architecture specifically addressed by the ATML family of standards.
The term "ATML family of standards"	Represents the complete set of ATML family component standards and associated XML schemas (see Table 3).
The term "ATML family component standard"	Represents a particular IEEE 1671 series standard (IEEE Std 1671.1™ [B31] through IEEE Std 1671.6™ [B36]).
The term "ATML common element schemas"	Reflects only the ATML Common, ATML HardwareCommon, and ATML TestEquipment XML schemas defined in this document (B.1, B.2, and B.3). These schemas **shall not have** associated XML instance documents.
The term "ATML internal model schemas"	Reflects only the ATML Capabilities and ATML WireLists XML schemas defined in this document (C.1 and C.2). These schemas **shall have** associated XML instance documents.
The term "subdomain"	Represents the complete set of ATML family component standards and associated XML schemas.
The term "subframework"	Represents the ATS or support software.
The term "external interface"	Represents the IEEE 1671 series standards (IEEE Std 1671.1 through IEEE Std 1671.6).
The term "internal model"	Represents the Capabilities and WireLists XML schemas defined in Annex C.
The term "ATML <component name> XML schema"	Represents the XML schema associated with the ATML family component standard. <component name> is defined in Table 3.
The term "ATML <component name> document"	Represents an XML instance document conforming to the ATML <component name> XML schema. For example, an ATML Test Description document is an XML instance document conforming to the ATML Test Description XML schema. <component name> is defined in Table 3.

[3] Information on references can be found in Clause 2.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

Annex A, Annex B, Annex C, Annex D, Annex E, Annex F, Annex G, and Annex I present XML schema and XML instance document information. The conventions used in their presentation are defined in Table 2.

Table 2—XML schema and XML instance document conventions

Item	Convention
All specifications in the XML language	Are given in the `Courier` type font where the XML elements are represented outside the XML schema or instance document.
The terms "isRef (0)" and "isRef (1)"	Is a XML boolean indicator used to identify whether an object is a reference. The term isRef (0) indicates it is not a reference; isRef (1) indicates that it is a reference.
The term "final #all"	Is an XML property that prevents all derivation. Used by the Complex Type `c:Extension`.
The use of "—" in tables	Indicates that no information is associated with this table cell or, with respect to attribute usage, implies optional.
The term "content simple"	Indicates that the XML element is not allowed to have attributes or subelements.
The term "content complex"	Indicates that a new complex data type is being defined, which can be used to declare elements to accept attributes and/or subelements.
The use of *italics*	Represents a XML element defined outside the subclause.
The use of "1 … ∞" and "0 … ∞" in tables	Represents the number of times an XML element may appear in an XML instance document. i.e., either one to infinity times or zero to infinity times.
XML snippets of XML instance documents	Are given in the `Courier` type font.
The XML attribute "`xsi:type`"	Explicitly declares the XML element type.
The use of "|" in XML simple types descriptions	Indicates a logical OR.

This document uses the vocabulary and definitions of relevant IEEE standards. In case of conflict of definitions, except for the portions quoted from standards, the following precedence shall be observed: (1) Clause 3, (2) Annex K, and (3) *The IEEE Standards Dictionary: Glossary of Terms & Definitions.*

For clarity, portions of IEEE Std 1641 [B29] have been duplicated within this document. In the event of revision to IEEE Std 1641, the current, approved version of the revised standard shall take precedence.

1.5.2 Word usage

In this document, the word *shall* is used to indicate a mandatory requirement. The word *should* is used to indicate a recommendation. The word *may* is used to indicate a permissible action. The word *can* is used for statements of possibility and capability.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). The latest edition of the referenced document (including any amendments, corrigenda, and/or working group drafts) applies unless the specific year of publication or edition is referenced.

Extensible Markup Language (XML) 1.0, (Fifth Edition). W3C Proposed Edited Recommendation, 5 Feb. 2008.[4]

[4] This document is available from the World Wide Web Consortium (W3C®) (http://www.w3.org/xml).

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. *The IEEE Standards Dictionary: Glossary of Terms & Definitions* should be referenced for terms not defined in this clause.[5] In the event a term is explicitly redefined or further defined in an ATML family component standard, that component standard's definition shall be normative only for that particular component standard.

abstract data type: A declared type that can be used to define other types through derivation. Only nonabstract types derived from the declared type can be used in instance documents. When such a type is used, it must be identified by the `xsi:type` attribute.

application: (A) The use to which a system is put. **(B)** The use of capabilities provided for by a system specific to the satisfaction of a set of users' requirements.

automatic test equipment (ATE): An integrated assembly of stimulus, measurement, and switching components under computer control that is capable of processing software routines designed specifically to test a particular item or group of items. ATE software includes operating system software, test executive software, and instrument control software.

NOTE—Definition adapted from DoD ATS Selection Process Guide [B7].[6]

automatic test equipment (ATE) control software: Software used during execution of a test program (TP) that controls the nontesting operations of the ATE. This software executes a test procedure but does not contain any of the stimuli or measurement parameters used in testing the unit under test (UUT). Where test software and control software are combined in one inseparable program, that program will be treated as test software, not control software.

NOTE—Definition adapted from MIL-STD-1309D [B52].

automatic test equipment (ATE) oriented language: A computer language used to program an ATE to test units under test (UUTs). The characteristics of this language imply the use of a specific ATE system or family of ATE systems.

NOTE—Definition adapted from MIL-STD-1309D [B52].

automatic test equipment (ATE) support software: Computer programs that aid in preparing, analyzing, and maintaining unit under test (UUT) test programs (TPs). Examples are ATE compilers and translation and analysis programs.

NOTE—Definition adapted from MIL-STD-1309D [B52].

automatic test equipment (ATE) system software: The total software environment of the ATE including operating system, test executives, user interface, system self-test, and other software required to run test programs (TPs). This term does not include TPs for supported end items.

NOTE—Definition adapted from MIL-STD-1309D [B52].

[5] *The IEEE Standards Dictionary: Glossary of Terms & Definitions* is available at http://shop.ieee.org/.
[6] Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement the standard.

Copyright © 2011 IEEE. All rights reserved.
Documents Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

automatic test markup language (ATML) capabilities: An extensible markup language (XML) schema that provides for the mapping of tests to instruments (and test systems) in a way that makes it possible to tell whether a test system is able to execute a given test.

automatic test markup language (ATML) instance document: *See*: **instance document**.

automatic test markup language (ATML) namespace: A collection of names, identified by a uniform resource identifier (URI) reference.

NOTE—Definition adapted from *Namespaces in XML 1.0* [B43].

automatic test markup language (ATML) ports, pins, and connectors: The description of instruments, test systems and their capabilities at the instruments, test systems pins, respectively.

automatic test system (ATS): A fully integrated, computer-controlled suite of electronic test equipment hardware, software, documentation, and ancillary items designed to verify at any level of maintenance the functionality of unit under test (UUT) assemblies. An ATS combines the following three elements: automatic test equipment (ATE), test program set (TPS), and test environment.

NOTE—Definition adapted from DoD ATS Selection Process Guide [B7].

D connector: A type of connector so named because one side is shorter (with one less pin) than the other, giving a physical D-shape.

digital rights management: Access control technologies that may be utilized to impose limitations on the usage of digital content material that are not foreseen by the content provider. Within the context of automatic test markup language (ATML), the digital content materials are the associated XML schemas, and the content provider is the IEEE.

element: A bounded component of the logical structure of an extensible markup language (XML) document that has a type and that may have XML attributes and content.

NOTE—Definition adapted from *Extensible Markup Language (XML) 1.0*.

entity: Something that has a distinct separate existence.

extensible markup language (XML) attribute: Name-value pair associated with an XML element.

extensible markup language (XML) document: A data object that conforms to the XML requirements for being well-formed. In addition, the data object is valid if it additionally conforms to semantic rules of the XML schema.

extensible markup language (XML) schema: The definition of a class of XML document, typically expressed in terms of constraints on the structure and the content of documents of that class, above and beyond the basic syntax constraints imposed by XML itself.

extensible markup language (XML) style sheet: A description of how an XML document is to be presented on a computer screen or in print.

fault: Degradation in performance due to detuning, misadjustment, misalignment, or failure of part(s).

framework: A real or conceptual structure intended to serve as guidance for building something that expands this structure into something useful.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

global attribute: An attribute declaration that is a child of the xs:schema element. A global attribute can be applied to any element.

hypertext markup language (HTML) viewer: A software program that enables a human to read an HTML file in its native format.

instance document: An extensible markup language (XML) document that conforms to a particular XML schema.

isRef: A boolean indicator used to identify that an extensible markup language (XML) object is a reference.

object: An entity that consists of state and behavior. An object stores its states in fields (variables in some programming languages) and exposes its behavior through methods (functions in some programming languages).

signal: (A) An electrical impulse controlled or observed by a test resource. **(B)** A visual, audible, or other indication used to convey information.

software tool: A software program that aids in the development of other software programs.

stimulus: Any physical or electrical input applied to a test subject intended to produce a measurable response.

NOTE—Definition adapted from MIL-STD-1309D [B52].

test environment: A description of the automatic test system (ATS) architecture, programming and test specification languages, compiler, development tools, and provisions for capturing and using unit under test (UUT) design requirements and test strategy information in the generation and maintenance of test program set (TPS) software.

NOTE—Definition adapted from DoD ATS Selection Process Guide [B7].

test executive: A software application that controls the execution environment of unit under test (UUT) test programs (TPs). Typical functions include, but are not limited to, verifying hardware/software availability, interpreting and executing operators commands, initializing and controling tests, providing common subprograms for test software usage, providing debug and simulation capabilities, logging test data, and allocating virtual resources.

test program set (TPS): Automatic test equipment (ATE) interface hardware, test program (TP) software, documentation, and other ancillary equipment that connects the unit under test (UUT) to the ATE. The TPS software performs fault isolation and diagnostics and can certify a UUT as ready for issue. Ancillary hardware consists of probes, holding fixtures, and peculiar instrumentation.

NOTE—Definition adapted from DoD ATS Selection Process Guide [B7].

unicode: An industry standard designed to allow text and symbols from all languages to be consistently represented and manipulated.

unicode transformation format (UTF): A variable-length character encoding for unicode.

well-formed: Conforming to all of extensible markup language (XML) syntax rules.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

3.2 Acronyms and abbreviations

AI-ESTATE	Artificial Intelligence Exchange and Service Tie to All Test Environments
AM	amplitude modulation
API	application programming interface
ARB	arbitrary waveform generator
ATE	automatic test equipment
ATLAS	Abbreviated Test Language for All Systems
ATML	automatic test markup language
ATS	automatic test system
BIT	built-in test
BNC	Bayonet Neill Concelman connector
BSC	basic signal component
CAGE	commercial and government entity
CORBA	common object request broker architecture
COM	component object module
CSCI	computer software configuration item
DLL	dynamic link library
DMM	digital multimeter
FM	frequency modulation
GUID	globally unique identifier
HTML	hypertext markup language
IC	integrated circuit
ID	identifier or interface device
I/O	input/output
ITA	interface test adapter
IVI®	interchangeable virtual instrumentation
LAN	local area network
LRU	line replaceable unit
PC	personal computer
PCB	printed circuit board
PM	phase modulation
R&D	research and development
RF	radio frequency
RFI	receiver/fixture interface
rms	root mean square

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

SGML	standard generalized markup language
SCC20	Standards Coordinating Committee 20
SCPI	standard commands for programmable instrumentation
SI	synthetic instrumentation
SIMICA	Software Interface for Maintenance Information Collection and Analysis
SMA	subminiature A connector
SRA	shop replaceable assembly
SRU	shop replaceable unit
STD	signal and test definition
TAR	test accuracy ratio
TII	Test Information Integration
TP	test program
TPS	test program set
TSF	test signal framework
UML	unified modeling language
URI	uniform resource identifier
URN	uniform resource name
URL	universal resource locator
UTC	coordinated universal time
UTF-8	8-bit unicode transformation format
UUID	universal unique identifier
UUT	unit under test
VME	versa module europa
VPP	VXI plug and play
VSWR	voltage standing wave ratio
VXI	VMEbus extensions for instrumentation
W3C®	World Wide Web Consortium
WRA	weapons replaceable assembly
WSDL	web services definition language
XHTML	extensible hypertext markup language
XML	extensible markup language
XSD	XML schema document

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

4. Automatic test system (ATS) architecture[7]

ATSs are utilized to identify failed electronic components, adjust these components to meet specifications, and assure that an electronic system or electronic component is "ready for issue"; in other words, the item is functioning as it was designed to operate.

An ATS includes the following:

a) ATE hardware and its operating software.

b) TPSs, which include the hardware, software, and documentation required to interface with, and test, individual component items. The associated software development environments required to produce the TPS are also included.

c) Automatic diagnostics and testing.

4.1 Automatic test equipment (ATE)

ATE refers to the test hardware and its accompanying software.

ATE utilizes one or more computers to control test instruments such as digital voltmeters, waveform analyzers, signal generators, and switching assemblies. This equipment operates under control of test software to provide a stimulus to a particular circuit or component in the UUT and then measure the output at various pins, ports, or connections to determine whether the UUT has performed to its specifications. The basic definition of ATE, then, is computer-controlled stimulus and measurement.

ATE is widely used in the electronic manufacturing industry to test electronics components and systems, both before and after they are fabricated. These electronic components and systems include (but are not limited to) avionics systems on commercial and military aircraft, electronic modules in automobiles, and electronic medical devices.

An ATE can be configured to test:

a) Simple components (resistors, capacitors, and inductors).

b) Integrated circuits (ICs).

c) Printed circuit boards (PCBs). PCBs can also be referred to as either shop replaceable units (SRUs) or shop replaceable assemblies (SRAs).

d) Black boxes [sometimes called either line replaceable units (LRUs) or weapons replaceable assemblies (WRAs)].

e) "All Up Round" weapons and weapon sections.

f) Other related electronic components or modules.

4.1.1 ATE hardware

The hardware architecture of an ATS will depend on its planned use, e.g., research and development (R&D), design validation, manufacturing, or field support; on budget and development-time constraints; existing expertise; and measurement throughput requirements.

[7] Concepts described in this clause have, in part, been derived from the DOD ATS Handbook [B5].

Copyright © 2011 IEEE. All rights reserved.
Documents Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

In R&D, for example, parametric tests are performed but will not be repeated on hundreds of UUTs. In high-volume manufacturing, for example, hundreds to thousands of UUTs may be tested, and each has to be tested as fast and as inexpensively as possible.

The hardware architecture of the ATE will be different in each of these situations. Typically, the ATE hardware architecture contains six major subsystems:

a) Instrumentation (measuring and stimulus instruments)
b) Computing [computer, software, and input/output (I/O)]
c) Switching (relays that interconnect system instrumentation and loads to the UUT)
d) Mass interconnects (UUT-to-system wiring interface)
e) Power sources (power to the UUT)
f) UUT-specific connections (e.g., loads, serial interfaces)

Each of these six subsystems is depicted by the grey shaded items in Figure 1.

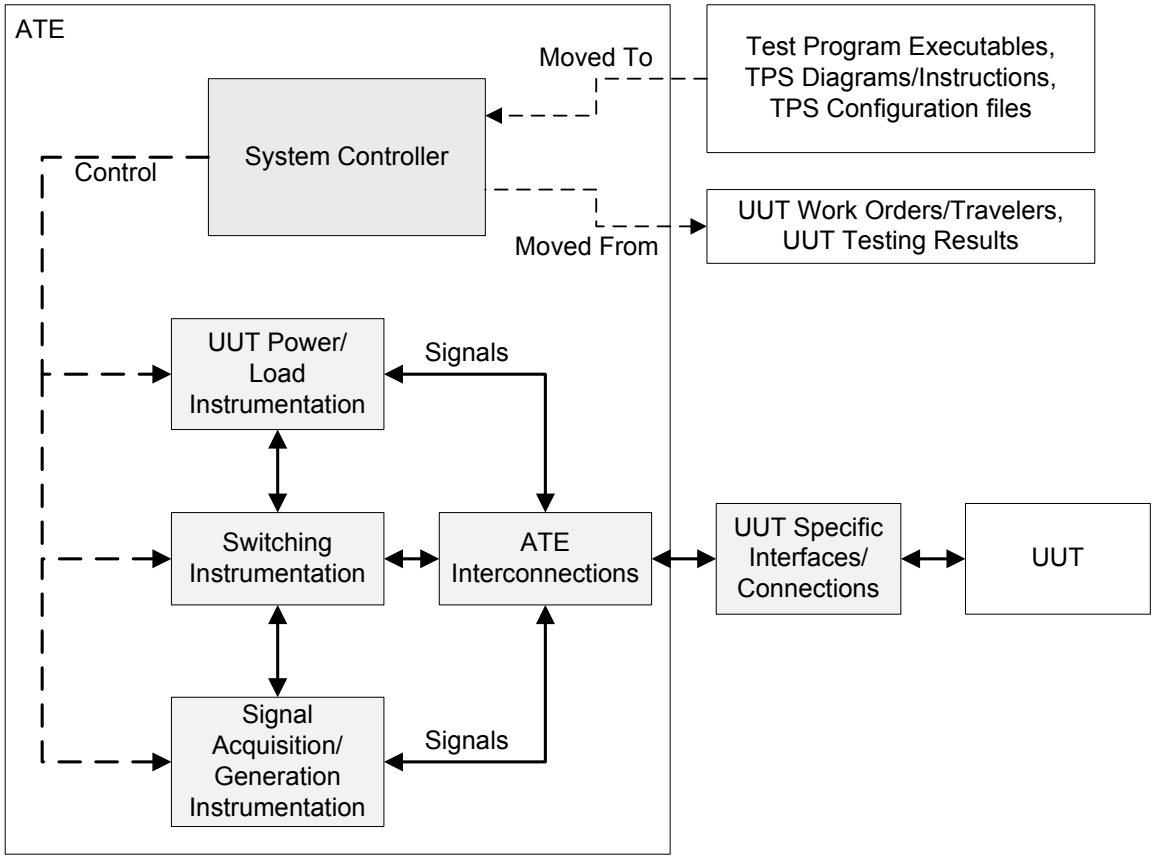


Figure 1—ATS hardware subsystems

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

4.1.2 ATE software

A "typical" ATE software architecture consists of at least the following two computer software configuration items (CSCIs):

a) ATE support software

b) ATE system software

ATE support software comprises software items typically running on a standalone personal computer (PC) for the purposes of developing a TP, while the ATE system software comprises software items running on the ATE system controller executing the TP. Both the ATE system software and the ATE system software are depicted in Figure 2 (as the "callouts" over the ATS depicted in Figure 1).

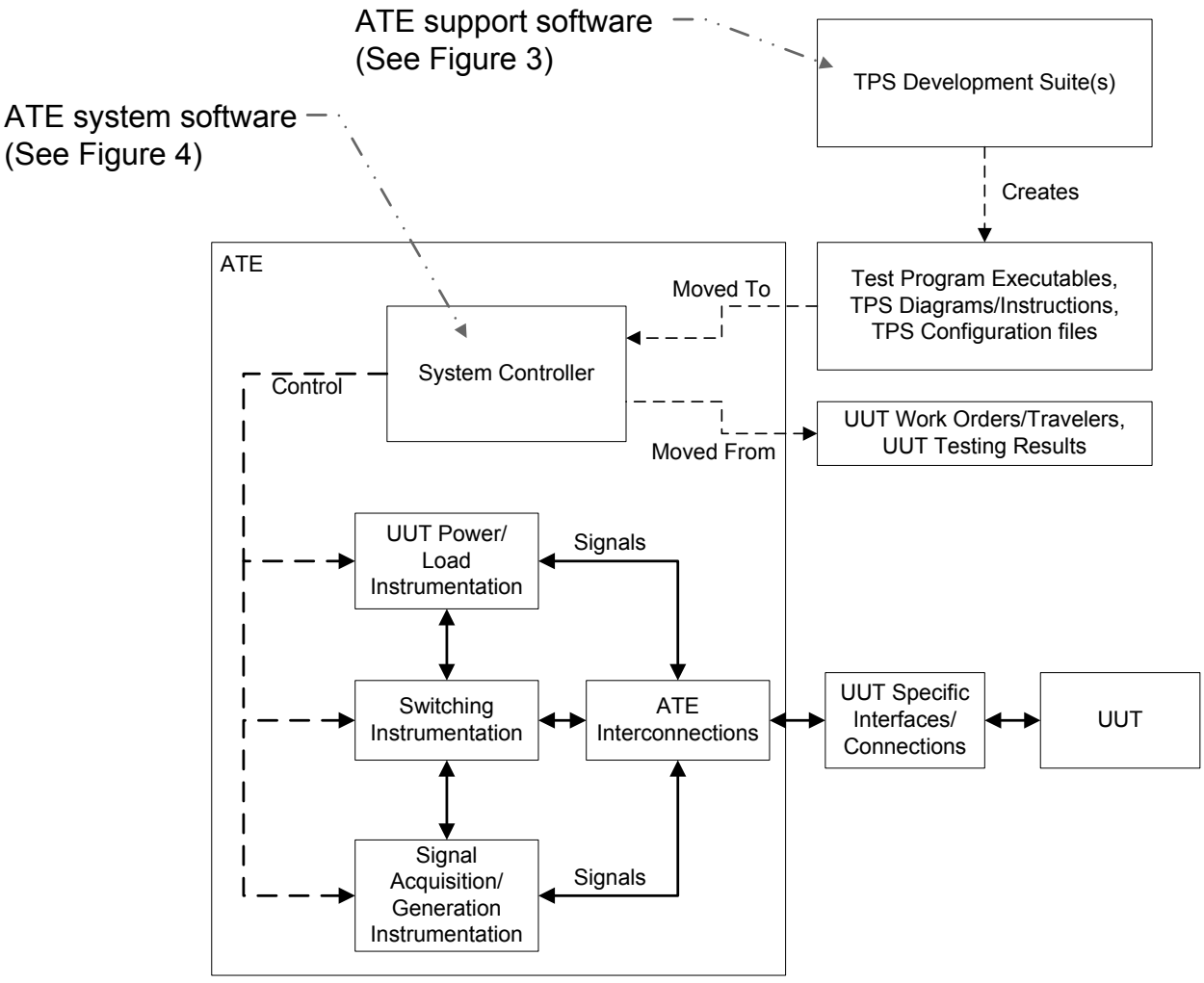


Figure 2—Software associated with a "typical" ATS

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

4.1.2.1 ATE support software

ATE support software consists of the software that aids in the preparation, analysis, and maintenance of UUT TPs. The ATE support software typically is not available on the ATE station.

In order to actually conduct tests on the UUT utilizing the ATE and interface test adapter (ITA), a UUT TP needs to be developed from the UUT testing strategy so that it may be executed by the ATE station control software.

Historically, the elements of a particular ATE's support software incorporated interpretations by the ATE developers of such items as instrument vendor data sheets/documentation, ATE hardware design material, etc. These interpretations are effectively turning one data format into a second (usually proprietary) format (e.g., an instrument data sheet contents in PDF format, put into a compiler's instrument database's unique file format). This usually always loses something in the translation as information is lost, does not have a home, etc.

"Typical" ATE support software items are depicted in Figure 3 (as the "TPS Development Suite(s)" over the ATS depicted in Figure 1).

4.1.2.2 ATE system software

ATE system software is the total software environment of the ATE including operating system, test executives, user interface, system self-test, and other software required to run UUT TPs.

In order to run the tests on the UUT, the UUT TP created via the ATE support software (see 4.1.2.1) needs an environment to be executed from.

Historically, the elements of a particular ATEs station control software interfaced with and produced data in proprietary formats (TP intermediate programming languages, test results, etc.) This meant that only the matching ATE support software could be utilized, and that test results were represented/store in a format unique to that ATE.

"Typical" ATE system software items are depicted in Figure 4 (as the "System Controller Software" over the ATS depicted in Figure 1).

4.2 Test program set (TPS)

TPSs consist of the test software, interface devices, cabling, and associated documentation.

The computer(s) in the ATE execute the test software, which usually is written in a standard language such as Abbreviated Test Language for All Systems (ATLAS), C, or Visual Basic. The stimulus and measurement instruments in the ATS have the ability to respond as directed by the computer. They send signals where needed and take measurements at the appropriate points. The test software then analyzes the results of the measurements and determines the probable cause of failure. It displays to the technician the component to remove and replace.

Developing the test software requires a series of tools collectively referred to as the software development environment. These tools include ATE and UUT simulators, ATE and UUT description languages, and programming tools such as compilers.

Since each UUT likely has different connections and I/O ports, interfacing the UUT to the ATE normally requires an interconnecting device (known as an ITA) and cables, which physically connect the UUT to the ATE and route signals from the various I/O pins in the ATE to the appropriate I/O pins in the UUT.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

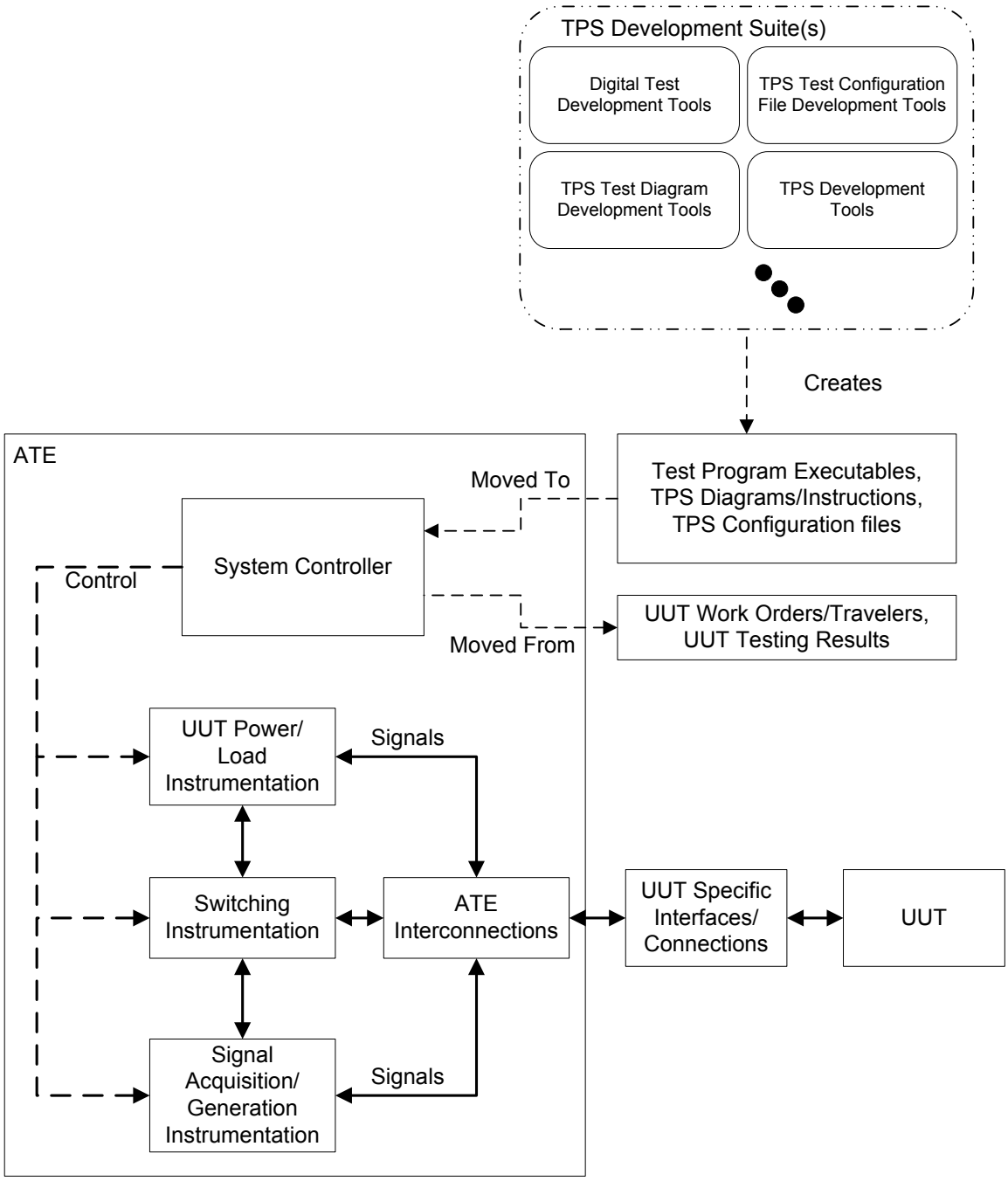


Figure 3—ATE support software

Copyright © 2011 IEEE. All rights reserved.
Documents Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

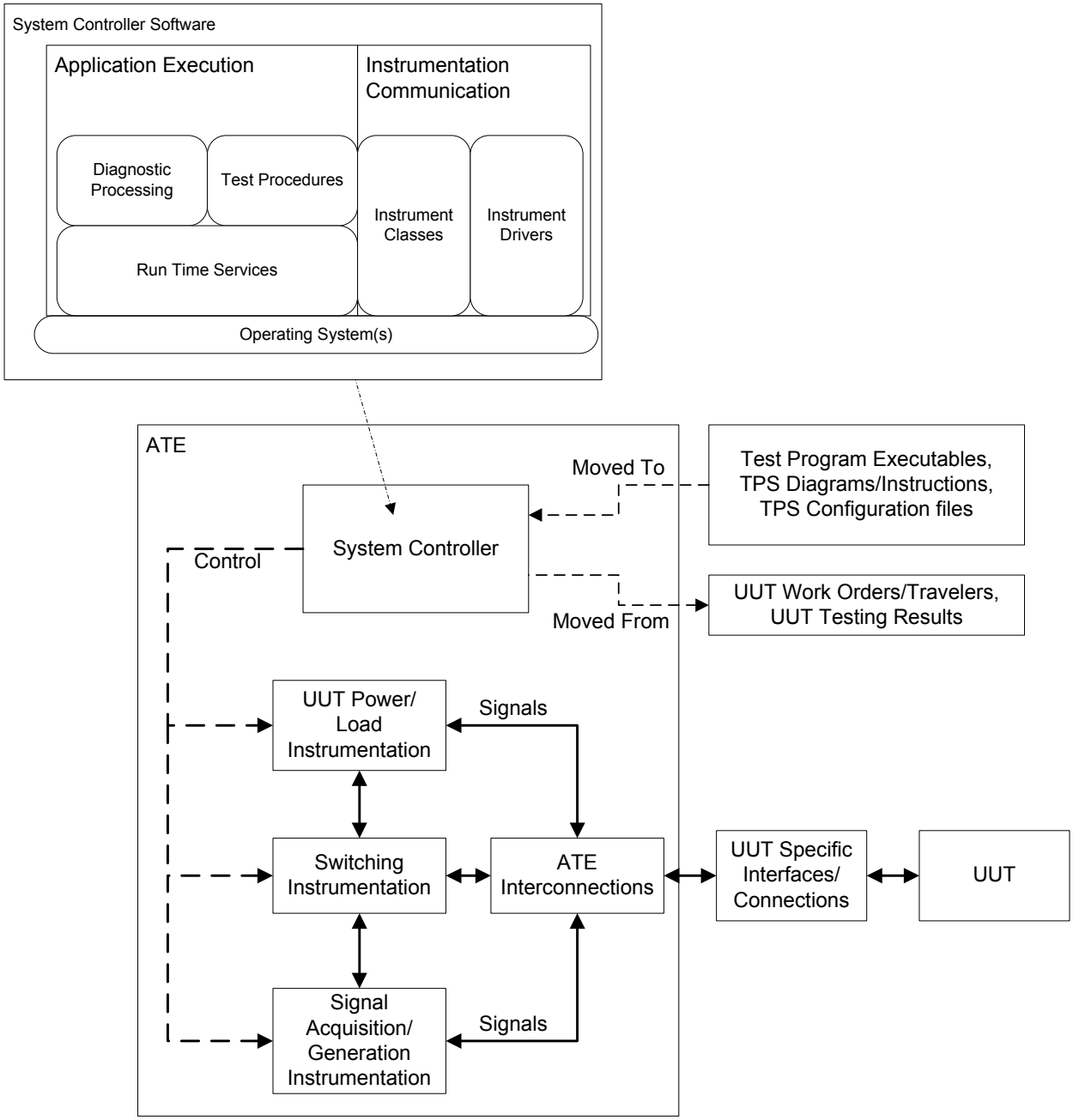


Figure 4—ATE system software

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

4.3 Automatic diagnosis and testing

Diagnostics is the part of an ATE test that determines the faulty components. ATE tests perform two basic functions. The first is to test whether the UUT is working correctly. The second is to diagnose the reason the UUT is not working correctly. The diagnostic portion can be the most difficult and costly portion of the test. It is typical for ATE to reduce a failure to a cluster or ambiguity group of components.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

5. Automatic test markup language (ATML)

The ATML initiative has come about by the desire to standardize on the XML format, rather than the various proprietary tools and formats used within the test industry (as discussed in Clause 4). By using a common format, different tools and systems can exchange information and be brought together to form cooperative heterogeneous systems, which, through the use of ATML, can

- Decrease test times.
- Reduce incidents of **Can Not Duplicate** or **No Fault Found.**
- Reduce the repair cycle.
- Formalize the capture of historic data that have been the preserver of experts in the field to heuristically identify faulty components.
- Close the loop on diagnostic systems.

The goals of 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 general uses cases ATML supports are to

- Support 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 TP 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 and maintenance information.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

5.1 ATS architecture elements addressed by ATML

ATML formally standardizes the following distinct subdomains under the ATML framework:

- The descriptions of how a test signal description "maps" to a ATE station (e.g., capabilities).
- The descriptions of test instrumentation.
- The descriptions of ITAs.
- The descriptions of the ATS configurations under which a UUT can be tested.
- The descriptions of UUT tests.
- The descriptions of test stations.
- The descriptions of a UUT.
- The descriptions of the elecrical paths from the UUT to the instrument in the ATE, on a per-test basis.

These sub-domains are depicted as the "callouts" in Figure 5, which are depicted "over the top" of the ATS architecture described in Clause 4.

A complete ATS architecture, including these ATML framework subdomains, is provided by Annex H.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

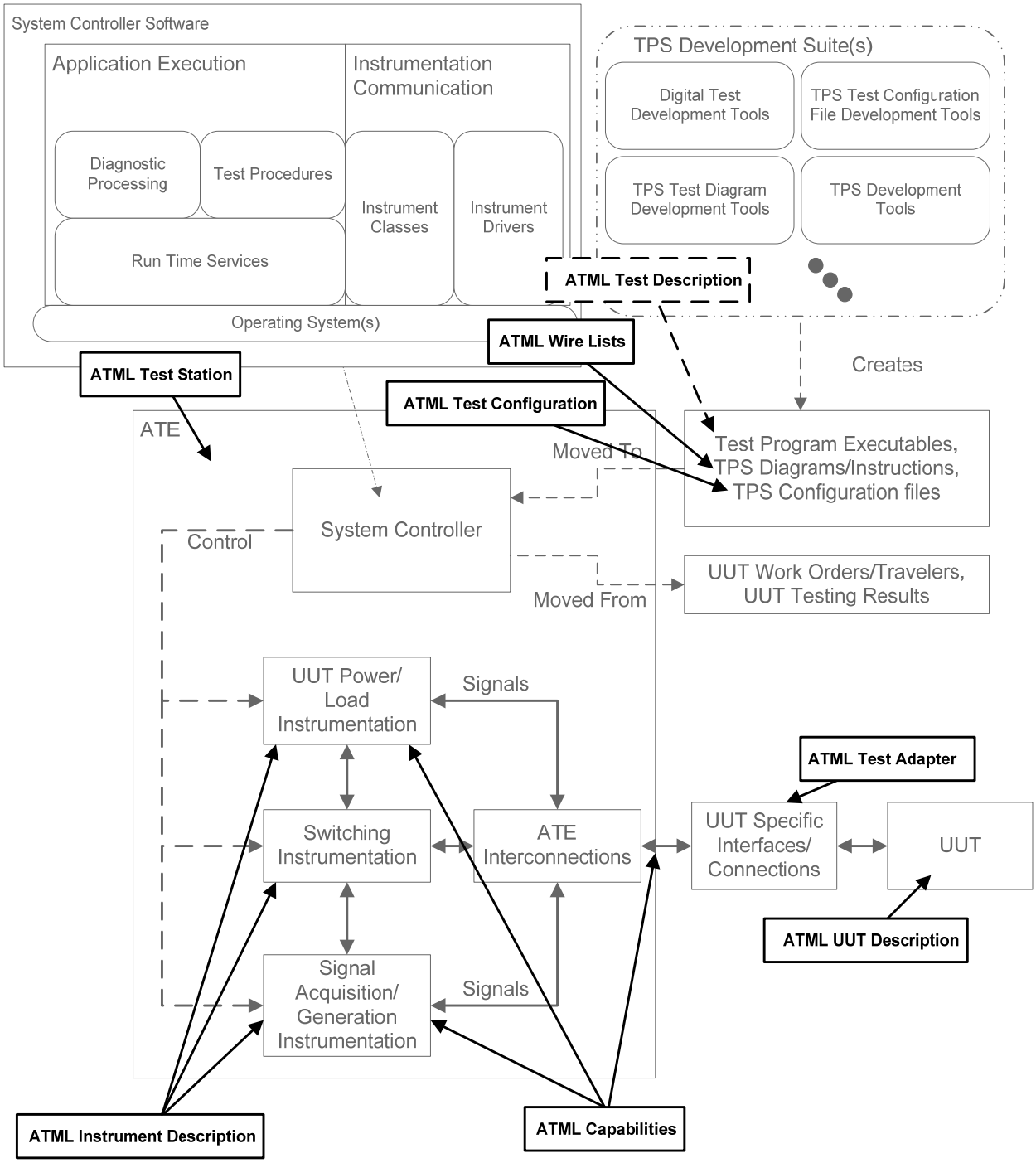


Figure 5—The ATML family component standards associated with a "typical" ATS architecture

NOTE—The ATML Test Description in Figure 5 is represented as a "dashed line" because the Test Program Executables are to be "derived from" the ATML Test Description.

Copyright © 2011 IEEE. All rights reserved.
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

6. The ATML framework

The ATML framework is based upon XML. XML describes a class of data objects called XML documents and partially describes the behavior of computer programs, which process them. XML is an application profile or restricted form of the standard generalized markup language (SGML) (see ISO 8879:1986 [B39]). By construction, XML documents are conforming SGML documents.

The ATML framework has 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)[8] standards.
- Be based upon standards.
- Be extensible.
- Enable the creation of modular ATS architectures (components based upon the ATML family component standards can easily be substituted, and data can be shared between the components).

The ATML framework is defined in the form of three distinct approaches:

- External interfaces
- Internal models
- Services

All external interfaces and internal models shall be with reference to the ATS and UUT. The ATML framework expects, although will not define, services to be available to generate, consume, and manipulate this information.

6.1 External interfaces

The external interfaces represent information that is exchanged between two or more distinct components in a typical ATS being used for testing of an UUT.

As defined in 5.1, and depicted by Figure 5 and Figure J.1, ATML formally standardizes six distinct external interfaces as ATML framework subdomains:

- Instrument descriptions
- Test adapter descriptions
- Test configurations

[8] This information is given for the convenience of users of this standard and does not constitute an endorsement by the IEEE of these consortium standards. Equivalent standards or products may be used if they can be shown to lead to the same results.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

— Test descriptions

— Test station descriptions

— UUT descriptions

Collectively, these six ATML framework subdomains were deemed to be the interfaces that offer the largest potential of reducing the cost to rehost or replace ATS components (up to as much as the entire ATE or the entire TPS).

NOTE—One with experience in ATE and/or TPSs might recognize that **a TP is not one of the ATML framework subdomains**. This omission is not an oversight. The ATML philosophy is that, since a TP is typically developed from test strategies and test requirements, the description of these test strategies and test requirements should be standardized as opposed to standardizing on the implementation of them (e.g., the TP). This approach allows for scenarios where the TPs are written in different languages. An example of this ATML philosophy is provided in I.2. This philosophy permits test strategies and test requirements to be implemented in the software programming language of choice, with the added benefit of potentially accessing a newer instrument's added capabilities in the future. ATML standardizes these test strategies and test requirements as part of the ATML Test Description component.

6.2 Internal models

Internal models ensure a consistent approach to defining elements that need common semantics. Within ATML there are several such models:

— Signal definitions using IEEE Std 1641 [B29]

— ATML capabilities

— ATML wire lists and network lists

As defined in 5.1, and depicted by Figure 5, ATML formally standardizes two distinct internal models as ATML framework subdomains:

— ATML capabilities

— ATML wire lists

The use of these items within the ATML framework ensures that different elements interpret the same information in the same way.

6.3 Services

The definition of the external interfaces (6.1) and internal models (6.2) is generally not enough to achieve interoperability. A simple scenario of **Tell me your test configuration?** or **What is the next test?** requires not only the definition of the format of the information, but also a definition of 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 the web services definition language (WSDL) that defines the format of messages the web

Copyright © 2011 IEEE. All rights reserved.
Documents Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

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.

ATML does not formally define specific services; however, ATML implementations should define services using WSDL. An example WSDL service is provided in Annex D.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

7. ATML specification techniques

The ATML framework is defined in terms of subframeworks and subdomains.

There are two major subframeworks, the ATS subframework and the support subframework.

The ATS sub-framework is analogous to the ATE system software defined in 4.1.2.2, and the support subframework is analogous to the ATE support software defined in 4.1.2.1.

The ATS subframework is further subdivided into following subframeworks:

— UUT & TPS
— Support software
— System software.

The support subframework is not subdivided.

Each of these four (i.e., UUT & TPS, support software, system software, and support) subframeworks contains one or more ATML subdomains. ATML subdomains are analogous to the following:

a) The six external interfaces described in 6.1.
    1) Each of the six are IEEE 1671 series ‘dot’ standards (e.g., IEEE Std 1671.1 [B31] through IEEE Std 1671.6 [B36], inclusive).
b) The two internal models described in 6.2.
    1) Both are defined in Annex C of this standard.
c) The three ATML common element schemas derived from the partitioning described in 7.1.
    1) Each of the three are defined in Annex B of this standard.

The ATML framework, the subframeworks, and the ATML subdomains are depicted by Figure 6.

7.1 ATML common element partitioning

Common elements provide for the definition of XML types and attributes that are utilized within more than one ATML subdomain XML schemas.

Common element XML schemas are reference XML schemas containing only type definitions that may be used in other XML schemas. **They have no root element, and there will be no instance documents directly validated against them.**

Having each ATML subdomain XML schema include ATML common elements allows for a consistent definition of shared XML types and prevents each XML schema from defining XML types used by other ATML subdomain XML schemas (which would have had to also define that XML type, possibly differently).

Common elements, as a result, is simply a **toolbox** for the ATML subdomain XML schemas to include.

Copyright © 2011 IEEE. All rights reserved.
Documents Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

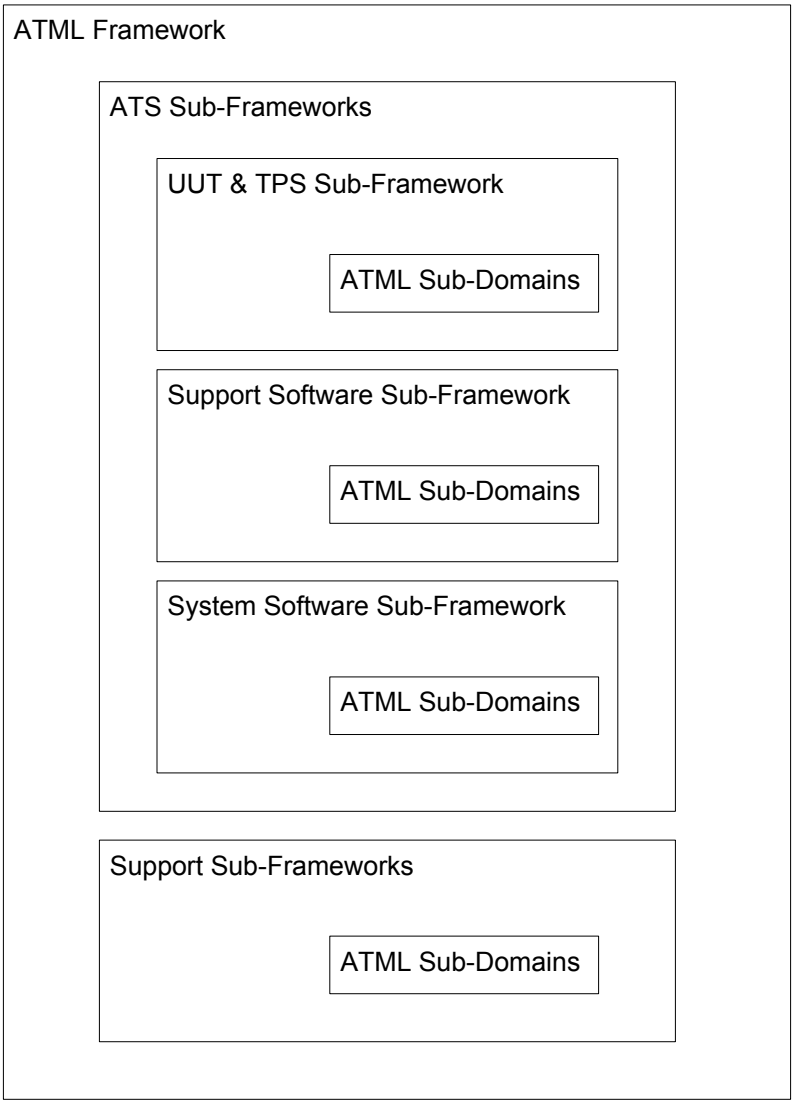


Figure 6—The ATML framework, subframeworks, and ATML subdomains

Figure 7 represents an example of a common element XML type (in this case, a NonBlankString) being inherited by two different XML schemas complex types (in this case, ExampleA and ExampleB), which are then used to develop two separate XML instance documents.

The fact that the XML schemas complex type's attribute is inherited is irrelevant to both the XML instance documents content and to how the XML instance document was or is generated.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

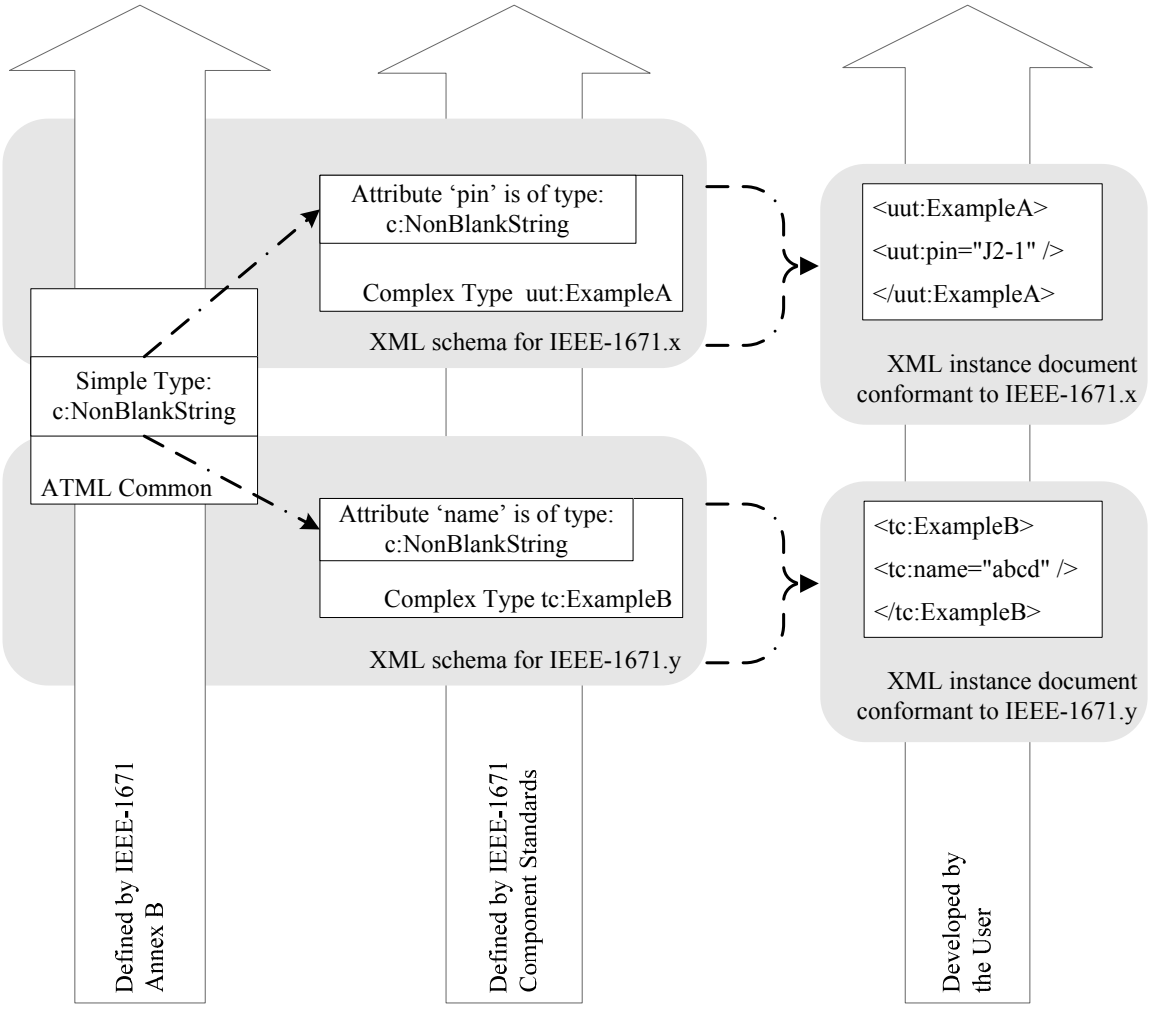


Figure 7—Example ATML common element usage

ATML defines three common element XML schemas:

a) **Common.xsd**, which provides ATML-unique types and attributes.

b) **HardwareCommon.xsd**, which provides ATML-hardware-unique types and attributes as well as includes Common.xsd.

c) **TestEquipment.xsd**, which provides test-equipment-unique types and attributes as well as includes both Common.xsd and HardwareCommon.xsd.

These three common element XML schemas are defined in B.1, B.2, and B.3, respectively.

Every ATML subdomain component XML schema (with the exception of Common.xsd) shall include at least one of the three common element XML schemas.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

7.2 ATML XML schemas

ATML defines a collection of XML schemas that allow ATE and test information to be exchanged in a common format adhering to the XML standard. These XML schemas have been developed, and continue to be maintained, through the use of modeling tools and use cases.

NOTE—It is expected that ATML users will utilize one or more XML-based tools to aid in the development of XML instance documents and/or to graphically view the XML schemas.

7.3 XML schemas and their use in ATML

The XML language focuses on the definition of entities, which are the objects of interest. The entities are defined in terms of their elements and attributes, which are the traits or characteristics considered to be important for using and understanding the entities. These elements or attributes have a representation, which might be a simple data type (such as integer) or another entity type. The XML schema also specifies constraints, rules, and relationships between entities.

ATML uses XML schemas to precisely specify the data that can reside in an ATML framework. XML schemas shall be specified for the categories of test information where different sets of data can be instantiated and exchanged between ATML implementations (as depicted in Clause 7 and defined in Clause 8). Test information that conforms to the ATML family XML schemas should be accessed and manipulated by software tools in an ATML test environment. The guidelines for the development of ATML XML schemas are provided in Annex A.

NOTE—**Some constraints cannot be represented by an XML schema; consequently these constraints are specified in the corresponding ATML family standard's textual content (e.g., the published standard)**. Thus, the sources for the complete set of requirements are the ATML family of standards and their associated XML schemas. Moreover, **validation of instance documents against XML schemas does not guarantee that the instance documents satisfy ATML compliance requirements; additional compliance verification may be necessary** for constraints that are not expressed in the XML schema.

7.4 UML models

This document includes, in Annex J, informational unified modeling language (UML)[9] models. One represents a generic ATS testing an UUT (Figure J.1). The second (represented by Figure J.2, Figure J.3, Figure J.4, and Figure J.5) represents the relationship between components of an ATS and UUT.

[9] This information is given for the convenience of users of this standard and does not constitute an endorsement by the IEEE of these products. Equivalent products may be used if they can be shown to lead to the same results.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

8. The ATML framework subdomains

Each of the ATML subdomains' external interfaces (see 6.1) is formally defined by an ATML family component standard (e.g., IEEE 1671 series "dot" standard). The ATML subdomain internal models and common element schemas are defined within this document (see 6.2 and 7.1), which is also an ATML family component standard.

XML instance documents of these ATML family component standards make up the core elements of an ATML framework. Figure 8 portrays the ATML family of standards (which would actually be XML instance documents valid against the associated family component) making up a fully populated ATML framework.

8.1 The ATML framework and ATML family component standards

The ATML family component standards define the external interfaces (see 6.1), internal models (see 6.2), and common elements (see 7.1). ATML family component standards are segmented into the following:

a) The ATML family component standard document (in the form of a formal IEEE standard)

b) The ATML family component's associated XML schemas

c) References to ATML instance documents (however, specific instance documents shall not be part of the ATML framework)

8.1.1 ATML family component standards

The ATML family components each shall have an associated IEEE published standard. Each of these standards shall contain the definition, description, and use of each element of the ATML family component as well as define the conformance to that standard.

8.1.2 ATML family XML schemas

The ATML family of standards shall have associated XML schemas. XML schemas are described in 7.2.

These XML schemas shall be located on the World Wide Web at the locations specified in Clause 9.

8.1.3 ATML instance documents

ATML instance documents are a collection of specific information defined and organized by the referenced XML schema (e.g., a particular instrument's instance document shall contain the definition of the particular instrument, in accordance with the instrument description XML schema specification).

8.2 ATML subdomains

The ATML family of standards (along with their associated XML schemas) defines a logically related set of ATML information (e.g., ATML subdomain). These ATML family component standards elaborate on information that only appears as place holders in this document. The ATML family of standards is defined in Table 3.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

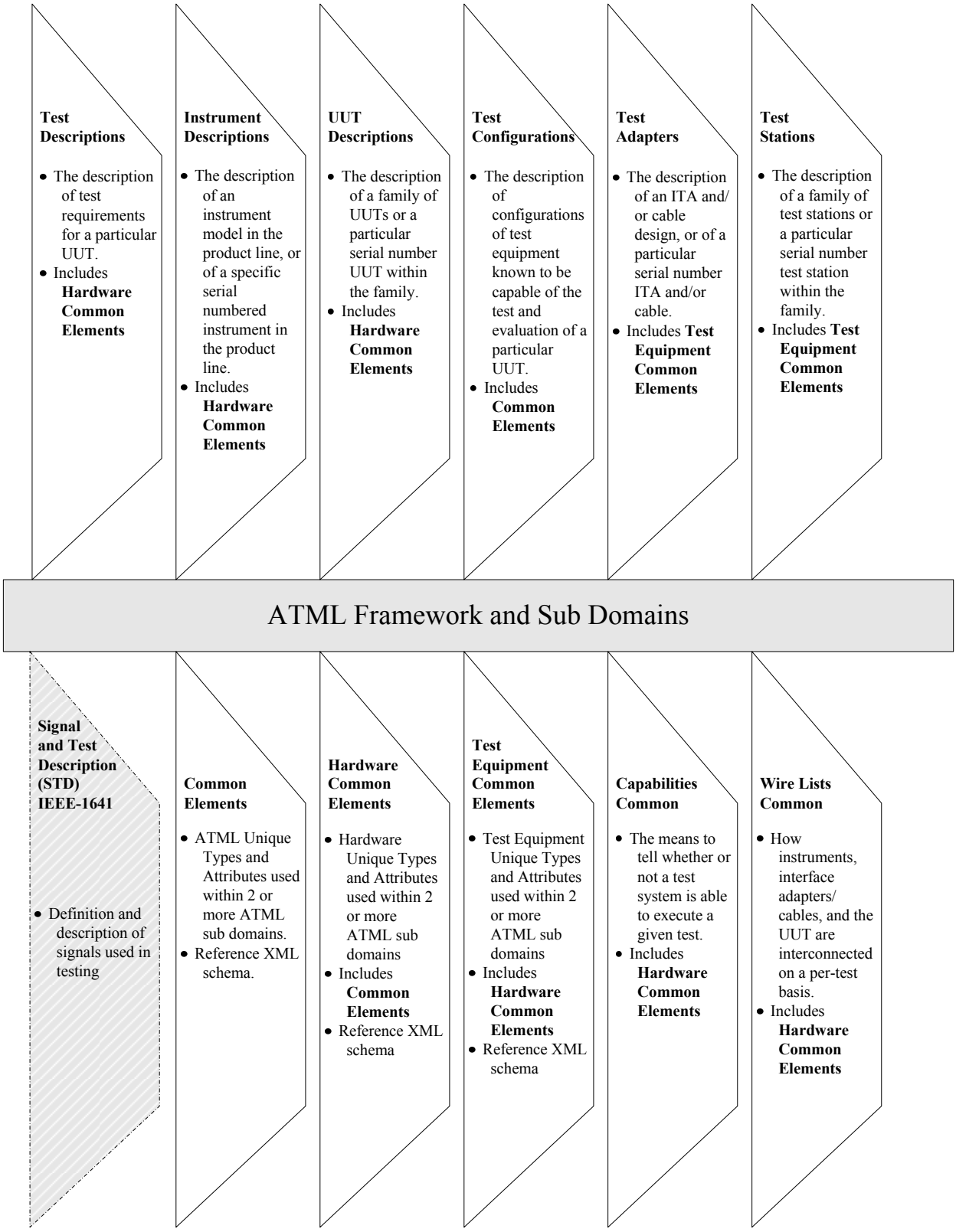


Figure 8—The ATML framework and subdomains

Copyright © 2011 IEEE. All rights reserved.
Documents Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

Table 3—ATML subdomains

Sub-domain name	Brief description	Standard
Standard Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML	This standard.	IEEE Std 1671 (ATML)
Capabilities	Annex F of this standard contains the necessary information to allow software to determine whether a given test system can run a given test. C.1 of this standard contains the Capabilities XML schema definition.	IEEE Std 1671 (ATML)
Common Elements	B.1, B.2, and B.3 of this standard contain the shared type definitions utilized within two or more ATML family components. There are three common element schemas: Common, HardwareCommon, and TestEquipment.	IEEE Std 1671 (ATML)
Instrument Description	Provides for the description of an test instrument.	IEEE Std 1671.2™ [B32] (ATML: Instrument Description)
Ports, Pins, Connectors, and Wire Lists	Annex E of this standard describes instruments, test systems, and their capabilities at the instruments' pins as "ports, pins connectors" and describes how ATS elements are interconnected as "wire lists." C.2 of this standard contains the WireLists XML schema definition.	IEEE Std 1671 (ATML)
Signal Definitions	Provides for the definition of a test.	IEEE Std 1641 [B29] (Signal and Test Definition)
Test Adapter	Provides for the description of an ITA and/or associated cables and other interface hardware.	IEEE Std 1671.5™ [B35] (ATML: Test Adapter)
Test Configuration	Provides for the description of the testing configuration.	IEEE Std 1671.4™ [B34] (ATML: Test Configuration)
Test Description	Provides for the description of the test subjects test requirements and a default test flow.	IEEE Std 1671.1 [B31] (ATML: Test Description)
Test Station	Provides for the description of a test station.	IEEE Std 1671.6 [B36] (ATML: Test Station)
UUT Description	Provides for the description of a test subject.	IEEE Std 1671.3™ [B33] (ATML: UUT Description)

8.2.1 Capabilities

An ATML framework may require a means to tell whether a test system is able to execute a given test.

ATML Capabilities provides a mechanism to allow tests to be mapped onto instruments (and test systems) in a way that makes it possible to tell whether a test system is able to execute a given test.

Any signal descriptions within a Capabilities description should be represented utilizing the signal and test definition (STD) standard (see IEEE Std 1641 [B29]).

ATML Capabilities is defined in Annex F. The Capabilities XML schema is defined in C.1.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

In the event ATML Capabilities is found to be insufficient or an error is identified, a change proposal to C.1 of this document should be directed to the Secretary, IEEE-SA Standards Board.

8.2.2 Common elements

Common elements provide for the definition of XML types and attributes that are utilized within more than one of the ATML family XML schemas. Common elements are described in 7.1.

In the event ATML Common is found to be insufficient or an error is identified, a change proposal to Annex B of this document should be directed to the Secretary, IEEE-SA Standards Board.

8.2.3 Instrument description

An ATML framework may require the description of an instrument model (e.g., a particular company's digital multimeter (DMM) model 123) or a specific occurrence of the instrument in the product line (e.g., a particular company's DMM model 123, serial number 1). Additionally, an ATML framework implementation may include a synthetic arbitrary waveform generator (ARB), synthetic digitizer, synthetic up-converter, or synthetic down-converter. Synthetic instrumentation (SI) requires descriptions like any other instrument.

The Instrument Description standard defines an exchange format for the static description of an nstrument. Instances of Instrument Description will be utilized in conjunction with instances of other Instrument Description in support of the execution of test programs in an automatic test environment. The standard promotes and facilitates interoperability between components of automatic test systems (e.g., between a test executive and a resource allocator) where instrument descriptions need to be shared.

SI is part of IEEE Std 1671.2 [B32], as both an example of Instrument Description instances as well as to provide a definition of the necessary parameters/attributes to document an SI. Template instance documents shall be used by vendors developing/providing SI, as the basis for documenting the SI. The XML template instance document provides examples for each instrument vendor to follow.

Any signal descriptions within an Instrument Description should be represented utilizing STD (see IEEE Std-1641 [B29]).

In the event ATML Instrument Description is found to be insufficient or an error is identified, a change proposal to IEEE Std 1671.2 [B32] and/or Annex B of this document should be directed to the Secretary, IEEE-SA Standards Board.

8.2.4 Ports, pins, connectors, and wire lists

An ATML framework may require a means to describe instruments, test systems, and their capabilities at the instruments' pins. ATML ports, pins, and connectors provide an explanation of the techniques used to map the capabilities to the instruments' pins.

ATML ports, pins, and connectors are described in Annex E.

An ATML framework may require a means to describe how instruments, test systems, interface adapters, and UUTs are interconnected. ATML wire lists provide an explanation of the techniques used to describe these interconnections. The interconnections are typically defined on a test-by-test basis, the test being an ATML Test Description document definition. Therefore, there shall be a direct mapping between wire list and test description.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

ATML wire lists are described in Annex E. The WireLists XML schema is defined in C.2.

In the event ATML WireLists is found to be insufficient or an error is identified, a change proposal to C.2 of this document should be directed to the Secretary, IEEE-SA Standards Board.

8.2.5 Signal definitions

IEEE Std 1641 [B29] is a culmination of a radical review of the ATLAS test programming language and the requirement to create truly portable test requirements. STD allows test information to pass more freely between the design, test, and maintenance phases of a project and enables the same information to be used directly across project phases. This more efficient use of information will lead to reduced life-cycle costs. STD provides the capability to describe and control signals, while permitting a choice of operating environment, including the choice of carrier language. STD permits signal operations to be embedded in any object-oriented environment and thus to be used by the architecture standards of various ATSs. Portability is extended beyond that of test specifications by virtue of a layered architecture. STD defines a collection of objects and their interfaces. These objects describe signal components relevant to test requirements. The STD standard defines how to interconnect these objects using interfaces through which the objects exchange information so that a test model may be defined that describes actual test requirements. Finally, the link to published ATLAS standards (such as IEEE Std 716™-1995 [B12]) is preserved in that the user can describe signal operations using very similar test-signal-related keywords. These keywords now have formal definitions. Furthermore, the parameters of the signals themselves also have a rigorous formal behavioral description.

Signal definitions are defined by IEEE Std 1641 [B29].

8.2.6 Test adapter

An ATML framework may require the description of an ITA and/or cable design or of a particular serial number ITA and/or cable. ITAs (also sometimes referred to as interface devices or IDs) are the interface between the UUT and the Test Station. Cables are the interface between the UUT, the Test Station, and ITA. What typically needs to be documented is the physical and electrical characteristics, the capabilities/performance, the identification and classification, etc. This information includes the connectors, wires, contacts, etc.

IEEE Std 1671.5 [B35] defines an exchange format for exchanging the test adapter information by defining the interface between the UUT and the test station, which includes the description of the test adapter, test adapter and/or ancillary cables[10], and any ancillary equipment required to interface the UUT to the ATE, in order to perform any test(s) on the subject UUT. These descriptions include the physical and electrical characteristics, capabilities/performance, and identification/classification. The standard provides a standardized format to promote and facilitate interoperability between components of automatic test systems by allowing the exchange of test adapter information. Each instance document contains the definition of a single test adapter or cable model. The test adapter schemas provide a structure for describing test adapter capabilities and structure.

Any signal descriptions within a Test Adapter description should be represented utilizing STD (see IEEE Std 1641 [B29]).

In the event ATML Test Adapter Description is found to be insufficient or an error is identified, a change proposal to IEEE Std 1671.5 [B35] and/or Annex B of this document should be directed to the Secretary, IEEE-SA Standards Board.

[10] When utilizing the Test Adapter XML schema for the description of any cable(s), the length of the cable is represented by depth.

Copyright © 2011 IEEE. All rights reserved.
Documents Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

8.2.7 Test configuration

An ATML framework may require the description of configurations of test equipment known to be capable of the test and evaluation of a particular UUT.

IEEE Std 1671.4 [B34] defines an exchange format for identifying all of the hardware, software, and documentation that may be used to test and diagnose a UUT on an ATS. The data support the acquisition and itemization of test assets required to be in place prior to testing a UUT on the test system.

In the event ATML Test Configuration is found to be insufficient or an error is identified, a change proposal to IEEE Std 1671.4 [B34] and/or Annex B of this document should be directed to the Secretary, IEEE-SA Standards Board.

8.2.8 Test description

An ATML framework may require the description of tests requirements for a particular UUT.

IEEE Std 1671.1 [B31] defines an exchange format for exchanging the test description information defining test performance, test conditions, diagnostic requirements, and support equipment to locate, align, and verify the proper operation of a UUT. This information shall be utilized in the development of TPSs that will be ultimately used in an automatic test environment. The standard promotes and facilitates interoperability between components of ATSs (e.g., rehosting test requirements between ATS platforms) where UUT test requirement definitions need to be shared.

Any signal descriptions within a Test Description should be represented utilizing STD (see IEEE Std 1641 [B29]).

In the event ATML Test Description is found to be insufficient or an error is identified, a change proposal to IEEE Std 1671.1 [B31] and/or Annex B of this document should be directed to the Secretary, IEEE-SA Standards Board.

8.2.9 Test station

An ATML framework may require the description a family of test stations or a particular test station within the family (e.g., by serial number). This description includes the physical and electrical characteristics; the paths between test system ports and the instrument; tolerances and accuracy of the test station; test station identification information such as part number, serial number, nomenclature, location; status information such as calibration data, dates, and self-test status; operational history, such as power-on time; external interfaces; power requirements; controller definitions; etc.

IEEE Std 1671.6 [B36] defines an exchange format for exchanging the test station information by defining the description of the test station (e.g., physical and electrical characteristics, components, capabilities/performance, identification/classification). The standard provides a standardized format to promote and facilitate interoperability between components of automatic test systems by allowing exchange of test station information. Each instance document contains the definition of a single test station model. The Test Station XML schema provides a structure for describing test station capabilities and structure.

Any signal descriptions within a Test Station description should be represented utilizing STD (see IEEE Std 1641 [B29]).

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

In the event ATML Test Station is found to be insufficient or an error is identified, a change proposal to IEEE Std 1671.6 [B36] and/or Annex B of this document should be directed to the Secretary, IEEE-SA Standards Board.

8.2.10 UUT description

An ATML framework may require the description of a family of UUTs or a particular UUT within the family (e.g., by serial number). This description includes information such as the name, part number, model number, type, power requirements, interfaces, physical properties, and operational requirements, i.e., the information about a UUT that is required to implement and execute tests on and diagnose the UUT itself.

IEEE Std 1671.3 [B33] defines an exchange format for information that uniquely describes a category or type of UUT. The format will include the ability to specify multiple manufacturers for each UUT, as there may be cases where a single UUT is supplied by a variety of manufacturers. This information is intended to support all aspects of the test and maintenance environment. The standard promotes and facilitates interoperability between components of test and maintenance support systems by defining a common set of identification information for UUTs.

In the event ATML UUT Description is found to be insufficient or an error is identified, a change proposal to IEEE Std 1671.3 [B33] and/or Annex B of this document should be directed to the Secretary, IEEE-SA Standards Board.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

9. ATML XML schema names and locations

The IEEE provides a download Web site for material associated with published IEEE standards; the material is presented in machine-friendly format. This material is digital rights management restricted use material. The ATML family of standards utilizes this download Web site to allow easy accessibility to all of the ATML family XML schemas (and in some cases, example XML instance documents). As depicted by Figure 9, the IEEE download Web site (http://standards.ieee.org/downloads/) contains several folders, each folder labeled by an associated IEEE standards number (e.g., IEEE 1671 series standards are in the 1671 folder). Each folder under this base IEEE standard number contains the material (e.g., XML schemas) for that ATML family component standard. ATML family component standards are identified by their IEEE 1671 series 'dot' standard number and the year in which that standard was published by the IEEE.

NOTE 1—Standards that are revised will contain a folder for the year in which the standard is reissued. Both folders (for each year the standard was published) will be present on the IEEE download Web site.
NOTE 2—Providing a particular standard has associated material that is to be made available via the download Web site, folders for that standard are not available until the standard is published by the IEEE.

Figure 9 depicts a portion of the entire IEEE download Web site, as it pertains to the ATML family of standards.

The IEEE SCC20 TII Subcommittee's Web site (http://grouper.ieee.org/groups/scc20/tii) provides access to material not yet published as an IEEE standard. This material is also digital rights management restricted use material. The ATML family of standards utilizes this site to allow easy accessibility to any of the ATML family XML schemas (and in some cases, example XML instance documents) not yet approved by the IEEE Standards Board.

The ATML family component standards (where the component is defined), their associated XML schemas' names, and the IEEE download Web site folder name (where the XML schemas shall be located) are as defined in Table 4.

Each of the XML schemas identified in Table 4 and Table 6 may include one or more of the ATML common element XML schemas defined in Annex A. The ATML common element (e.g., component) (where the component is defined), the associated XML schema's name, and the IEEE download Web site folder name (where the XML schema shall be located) are as defined in Table 5.

The XML schemas identified in Table 4 may utilize one or more of the ATML common XML schemas defined in Annex C. The ATML common schema (e.g., component) (where the component is defined), the associated XML schema's name, and the IEEE download Web site folder name (where the XML schema shall be located) are as defined in Table 6.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

http://standards.ieee.org/downloads/

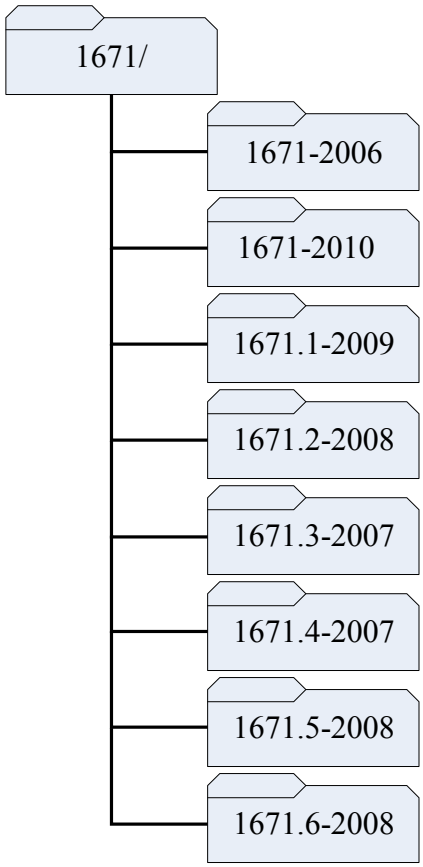


Figure 9—ATML-related IEEE download Web site structure

Table 4—ATML family XML schema names and folder locations

Component	Defined in	XML schema name	IEEE download Web site folder (see Figure 9)
Instrument Description	IEEE Std 1671.2	InstrumentDescription.xsd	1671.2-2008
		InstrumentInstance.xsd	1671.2-2008
		Digitizer.xml	1671.2-2008
		DownConverter.xml	1671.2-2008
		SyntheticWaveformGenerator.xml	1671.2-2008
		UpConverter.xml	1671.2-2008
Test Adapter	IEEE Std 1671.5	TestAdapterDescription.xsd	1671.5-2008
		TestAdapterInstance.xsd	1671.5-2008
Test Configuration	IEEE Std 1671.4	TestConfiguration.xsd	1671.4-2007
Test Description	IEEE Std 1671.1	TestDescription.xsd	1671.1-2009
Test Station	IEEE Std 1671.6	TestStationDescription.xsd	1671.6-2008
		TestStationInstance.xsd	1671.6-2008
UUT Description	IEEE Std 1671.3	UUTDescription.xsd	1671.3-2007
		UUTInstance.xsd	1671.3-2007

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

Table 5—ATML Common element XML schema names and locations

Component	Defined in	XML schema name	IEEE download Web site folder (see Figure 9)
Common	B.1	Common.xsd	1671-2010
Hardware Common	B.2	HardwareCommon.xsd	1671-2010
Test Equipment	B.3	TestEquipment.xsd	1671-2010

Table 6—ATML Common XML schema names and locations

Component	Defined in	XML schema name	IEEE download Web site folder (see Figure 9)
Capabilities	C.1	Capabilities.xsd	1671-2010
Wire Lists	C.2	WireLists.xsd	1671-2010

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

10. ATML XML schema extensibility

The provision of an extension mechanism is necessary to ensure the viability of the specification and allow producers and consumers of ATML instance documents to interoperate in cases where there is a requirement to exchange relevant data that are not included in the ATML family XML schemas. The use of the extensions shall be done in a way that ensures that a conformant consumer can utilize the extended file without error, discard or otherwise sidestep the extended data, and use the nonextended portions of the data as they are intended, without error or loss of functionality.

Extensions shall be additional information added to the content model of the element being extended.

Extensions shall not repackage existing information entities that are already supported by the ATML family of standards.

Extensions shall always be associated with a user-defined namespace and should be identified with a namespace prefix (see Table A.1).

An extended instance document shall be accompanied by the extension XML schema and documentation sufficient to explain the need for the extension as well as the underlying semantics and relationship(s) to the base ATML family XML schema.

The ATML family XML schemas allow for three forms of extension:

a) Wildcard-based extensions allow for the extension of the XML schemas with additional elements.

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

c) Lists derived from `c:NamedValues` allowing user-defined properties with attached values.

The ATML family XML schemas control the location and type of extension allowed. A.6.7 describes how to specify the extension points for an ATML family XML schema.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010

11. Conformance

ATML conformance has two facets. The first (11.1) shall apply only to the development and maintenance of the ATML family of standards and their associated XML schemas. The second (11.2) shall apply only to implementers and implementations of ATML (this standard).

11.1 ATML family XML schemas

Each of the ATML family XML schemas shall be developed and maintained to be conformant with the XML schema style guidelines defined in Annex A.

11.2 The ATML framework

Conformance to the ATML framework shall be achieved as specified by the requirements defined by either of the following:

a) Any combination of Table 7, and/or Table 8, and/or Table 9 (see 11.2.1 through 11.2.1.3)

b) Table 10 (see 11.2.2)

For each ATML family component standard utilized in either item a) or item b), the conformance requirements of the utilized component standard shall also be satisfied.

11.2.1 ATS subframework

Figure 10 illustrates three ATS subframeworks that may be incorporated within an ATS:

a) UUT & TPS (see 11.2.1.1)

b) ATE support software (see 11.2.1.2)

c) ATE System software (see 11.2.1.3)

An ATS shall include at least one of these subframeworks and may include all three.

Each of the ATS subframework components utilize ATML Common element schemas (see 7.1 and Table 5). The same ATML Common element schemas should be used for each ATS framework component; however, different versions of Common element schemas are allowed. It is also allowable to edit ATML family XML schemas, solely for the purpose of permitting the use of the same Common element schema version, within an ATS framework.

Copyright © 2011 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 22:29:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1671-2010