



**IEEE**

**IEC 61636**

Edition 1.0 2016-11

# **INTERNATIONAL STANDARD**

**IEEE Std 1636™**

**Software Interface for Maintenance Information Collection and Analysis (SIMICA)**

*IECNORM.COM : Click to view the full PDF of IEC 61636:2016*





**THIS PUBLICATION IS COPYRIGHT PROTECTED**  
**Copyright © 2013 IEEE**

All rights reserved. IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Inc. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the IEC Central Office. Any questions about IEEE copyright should be addressed to the IEEE. Enquiries about obtaining additional rights to this publication and other information requests should be addressed to the IEC or your local IEC member National Committee.

IEC Central Office  
3, rue de Varembe  
CH-1211 Geneva 20  
Switzerland  
Tel.: +41 22 919 02 11  
Fax: +41 22 919 03 00  
[info@iec.ch](mailto:info@iec.ch)  
[www.iec.ch](http://www.iec.ch)

Institute of Electrical and Electronics Engineers, Inc.  
3 Park Avenue  
New York, NY 10016-5997  
United States of America  
[stds.info@ieee.org](mailto:stds.info@ieee.org)  
[www.ieee.org](http://www.ieee.org)

**About the IEC**

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

**About IEC publications**

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

**IEC Catalogue - [webstore.iec.ch/catalogue](http://webstore.iec.ch/catalogue)**

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

**IEC publications search - [www.iec.ch/searchpub](http://www.iec.ch/searchpub)**

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

**IEC Just Published - [webstore.iec.ch/justpublished](http://webstore.iec.ch/justpublished)**

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

**Electropedia - [www.electropedia.org](http://www.electropedia.org)**

The world's leading online dictionary of electronic and electrical terms containing 20 000 terms and definitions in English and French, with equivalent terms in 15 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

**IEC Glossary - [std.iec.ch/glossary](http://std.iec.ch/glossary)**

65 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

**IEC Customer Service Centre - [webstore.iec.ch/csc](http://webstore.iec.ch/csc)**

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: [csc@iec.ch](mailto:csc@iec.ch).

IECNORM.COM - Link to view the full PDF of IEC 61636:2016



**IEEE**

**IEC 61636**

Edition 1.0 2016-11

# **INTERNATIONAL STANDARD**

**IEEE Std 1636™**

---

**Software Interface for Maintenance Information Collection and Analysis (SIMICA)**

*IECNORM.COM : Click to view the full PDF of IEC 61636:2016*

INTERNATIONAL  
ELECTROTECHNICAL  
COMMISSION

---

ICS 25.040.01; 35.060

ISBN 978-2-8322-3686-4

**Warning! Make sure that you obtained this publication from an authorized distributor.**

## Contents

1. Overview .....	1
1.1 General .....	1
1.2 Scope .....	3
1.3 Purpose .....	3
1.4 Application .....	3
1.5 Conventions used in this document .....	3
2. Normative references.....	3
3. Definitions .....	4
4. Backgrounds.....	5
4.1 Diagnostic maturation.....	5
4.2 Relationship to IEEE Std 1232 (AI-ESTATE) .....	6
4.3 Relationship to IEEE 1636 (SIMICA) component standards .....	6
5. Information model.....	6
5.1 SIMICA_MODEL.....	7
6. Conformance .....	20
Annex A (informative) Bibliography .....	21
Annex B (informative) Overview of EXPRESS.....	22
B.1 Schema.....	22
B.2 Entity.....	22
B.3 Attribute .....	23
B.4 Type definition.....	24
B.5 Subtypes/supertypes.....	24
B.6 External schema references.....	25
B.7 Uniqueness constraints and WHERE clauses .....	26
B.8 Functions and procedures.....	27
Annex C (informative) IEEE list of participants.....	28

## SOFTWARE INTERFACE FOR MAINTENANCE INFORMATION COLLECTION AND ANALYSIS (SIMICA)

### FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as “IEC Publication(s)”). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation.

IEEE Standards documents are developed within IEEE Societies and Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. IEEE develops its standards through a consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of IEEE and serve without compensation. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards. Use of IEEE Standards documents is wholly voluntary. IEEE documents are made available for use subject to important notices and legal disclaimers (see <http://standards.ieee.org/IPR/disclaimers.html> for more information).

IEC collaborates closely with IEEE in accordance with conditions determined by agreement between the two organizations.

- 2) The formal decisions of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees. The formal decisions of IEEE on technical matters, once consensus within IEEE Societies and Standards Coordinating Committees has been reached, is determined by a balanced ballot of materially interested parties who indicate interest in reviewing the proposed standard. Final approval of the IEEE standards document is given by the IEEE Standards Association (IEEE-SA) Standards Board.
- 3) IEC/IEEE Publications have the form of recommendations for international use and are accepted by IEC National Committees/IEEE Societies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC/IEEE Publications is accurate, IEC or IEEE cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications (including IEC/IEEE Publications) transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC/IEEE Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC and IEEE do not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC and IEEE are not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or IEEE or their directors, employees, servants or agents including individual experts and members of technical committees and IEC National Committees, or volunteers of IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board, for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC/IEEE Publication or any other IEC or IEEE Publications.
- 8) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that implementation of this IEC/IEEE Publication may require use of material 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. IEC or IEEE shall not be held responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patent 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 non-discriminatory. 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.

International Standard IEC 61636/IEEE Std 1636 has been processed through IEC technical committee 91: Electronics assembly technology, under the IEC/IEEE Dual Logo Agreement.

The text of this standard is based on the following documents:

IEEE Std	FDIS	Report on voting
1636 (2009)	91/1359/FDIS	91/1370/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

The IEC Technical Committee and IEEE Technical Committee have decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IECNORM.COM : Click to view the full PDF of IEC 61636:2016

# IEEE Standard for Software Interface for Maintenance Information Collection and Analysis (SIMICA)

Sponsor

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

Approved 19 March 2009

**IEEE-SA Standards Board**

Approved as a Full-Use Standard 5 December 2013

**IEEE-SA Standards Board**

IECNORM.COM : Click to view the full PDF of IEC 61636:2016

**Abstract:** This document provides an implementation-independent specification for a software interface to information systems containing data pertinent to the diagnosis and maintenance of complex systems consisting of hardware, software, or any combination thereof. These interfaces will support service definitions for creating application programming interfaces (API) for the access, exchange, and analysis of historical diagnostic and maintenance information. This will address the pervasive need of organizations to assess the effectiveness of diagnostics for complex systems throughout the product life cycle. The use of formal information models will facilitate exchanging historical maintenance information between information systems and analysis tools. The models will facilitate creating open system software architectures for maturing system diagnostics.

**Keywords:** AI-ESTATE, Automated Test Markup Language (ATML), diagnostic maturation, IEEE 1636™, Maintenance Action Information, maintenance data, Software Interface for Maintenance Information Collection and Analysis (SIMICA), Test Results and Session Information

IECNORM.COM : Click to view the full PDF of IEC 61636:2016

## IEEE Introduction

This introduction is not part of IEEE Std 1636-2009, IEEE Standard for Software Interface for Maintenance Information Collection and Analysis (SIMICA).

The requirement for a specification for access and exchange of diagnostic and maintenance product information has arisen due to a pervasive need for the organizations who deliver complex systems to monitor the effectiveness of their product health management solutions in their customers' application domains. Accordingly, The IEEE SCC20 Diagnostic and Maintenance Control (DMC) subcommittee has undertaken the task of developing a standard which fulfills this need. It is envisioned that SIMICA will fulfill this need for all such consumers of diagnostic and maintenance data.

## Notice to users

## Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document 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.

## 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 documents 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-SA Website at <http://standards.ieee.org/index.html> or contact the IEEE at the address listed previously. For more information about the IEEE Standards Association or the IEEE standards development process, visit IEEE-SA Website at <http://standards.ieee.org/index.html>.

## Errata

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

## 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 by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at <http://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. 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 non-discriminatory. 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.

IECNORM.COM : Click to view the full PDF of IEC 61636:2016

# Software Interface for Maintenance Information Collection and Analysis (SIMICA)

**IMPORTANT NOTICE:** IEEE Standards documents are not intended to ensure safety, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

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

This Standard Software Interface for Maintenance Information Collection and Analysis (SIMICA) was developed by the Diagnostic and Maintenance Control Subcommittee of the IEEE Standards Coordinating Committee 20 on Test and Diagnosis for Electronic Systems (SCC20) to provide standard, unambiguous definitions of maintenance information semantics, interrelationships, and access services. This standard defines a formal conceptual information model to relate maintenance information across concrete information models. These models are related to the maturation of diagnostic systems and as such are directly related to IEEE Std 1232<sup>TM</sup><sup>1</sup>. However this standard, in conjunction with its component standards, can also be used independent of AI-ESTATE.

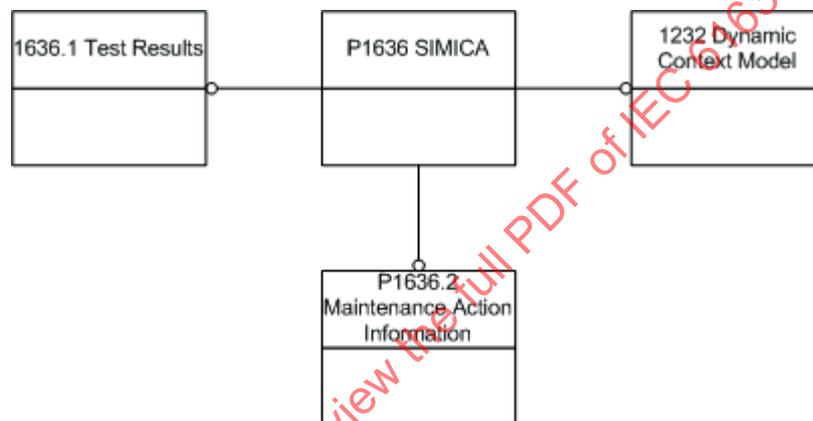
The goals of the 1636 family of standards are summarized here:

- Provide definitions of maintenance concepts and terminology relevant to the maturation of diagnostic systems.
- Provide a set of information models that will serve as a basis for unambiguous interpretation and communication of such data.

<sup>1</sup> Information on references can be found in Clause 2.

- Support the development of a software interface for moving such data between conforming applications.

This standard specifically describes a set of formal specifications consisting of the logical representation of the information that is used and generated during related diagnostic and maintenance processes. The information model contained in this document provides a standard model of the top-level information concepts that support these processes. Other components of this family of standards will further develop conceptual decompositions of key elements of these processes down to the elemental data level, where specific software interfaces can be represented. The relationship of these components to each other is shown in Figure 1. IEEE Std 1636.1™ defines an information model and XML schema for exchanging test result and session information. IEEE P1636.2™ defines an information model and XML schema for exchanging maintenance action information. Precise specification of the semantics of these data elements will provide the basis for services supporting the unambiguous exchange of information between producers and consumers of historical diagnosis and maintenance information in a platform-independent manner. SIMICA exhibits a close relationship to the IEEE Std 1232 through its direct use of Dynamic Context Model historical records of diagnostic sessions, which is also shown in Figure 1. The intent of this relationship is to relate diagnostic session information to associated maintenance actions for further analysis and maturation of diagnostic models and reasoner learning mechanisms.



**Figure 1—Relationship of SIMICA component EXPRESS schemata**

This standard family provides a controlled extension mechanism of its software interface components to allow inclusion of relevant new maintenance information elements currently outside the scope of the SIMICA specification.

The SIMICA family of standards defines key information specification formats. Implementations that use only these specification formats will be portable. This does not preclude use of SIMICA interfaces with non-conformant specification formats; however, such implementations may not be portable.

Software specifications defined in the SIMICA family of standards will support the interchangeability of information between conformant applications. This will allow a large degree of flexibility in implementation of information application architectures that support information reuse and realize various maintenance information services (such as, but not limited to, client-server approaches, service oriented architectures, etc.).

## 1.2 Scope

This standard is an implementation-independent specification for a software interface to information systems containing data pertinent to the diagnosis and maintenance of complex systems consisting of hardware, software, or any combination thereof. These interfaces will support service definitions for creating application programming interfaces (API) for the access, exchange, and analysis of historical diagnostic and maintenance information. The standard will use the information models of IEEE Std 1232 as a foundation.

## 1.3 Purpose

The purpose of this standard is to specify a software interface for access, exchange, and analysis of product diagnostic and maintenance information. This will address the pervasive need of organizations to assess the effectiveness of diagnostics for complex systems throughout the product life cycle. The use of formal information models will facilitate exchanging historical maintenance information between information systems and analysis tools. The models will facilitate creating open system software architectures for maturing system diagnostics.

## 1.4 Application

This standard should be applied in the development of software applications that access or provide information relevant to the diagnosis and maintenance of systems. In the event of conflict between this standard and a component standard in the IEEE 1636 family, the component standard shall take precedence. In the event of conflict between this standard and a related standard such as IEEE Std 1232, the standard as it applies to the information being produced shall take precedence. In the event of any conflict between model comments and lexical definitions, lexical definitions shall take precedence.

## 1.5 Conventions used in this document

Clause 5 presents entity and concept definitions using the EXPRESS language as defined in ISO 10303-11:1994 and uses the following conventions in their presentation:

All specifications in the EXPRESS language are given in the `Courier` type font when model elements are represented outside the scope of the model.

This standard uses the vocabulary and definitions of relevant IEEE standards. In case of conflict of definitions, the following precedence shall be observed: 1) Clause 3, Definitions; 2) *The Authoritative Dictionary of IEEE Standards Terms* [B1]<sup>2</sup> with preference given to versions of those definitions that are attributed to SCC20 standards.

## 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). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

---

<sup>2</sup> The numbers in brackets correspond to those of the bibliography in Annex A.

IEEE Std 1232™, IEEE Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE).<sup>3, 4</sup>

IEEE Std 1636.1™-2007, IEEE Standard for Software Interface for Maintenance Information Collection and Analysis (SIMICA): Exchanging Test Results and Session Information via the eXtensible Markup Language (XML).

IEEE P1636.2™/D3.0 (August 2008), Draft Trial-Use Standard for Software Interface for Maintenance Information Collection and Analysis (SIMICA): Exchanging Maintenance Action Information (MAI) via the eXtensible Markup Language (XML).<sup>5</sup>

ISO 10303-11:1994. Industrial automation systems and integration—Product data representation and Exchange—Part 11: Description methods: The EXPRESS language reference manual, Geneva, Switzerland: International Organization for Standardization.<sup>6</sup>

### 3. Definitions

For the purposes of this standard, the following terms and definitions apply. *The Authoritative Dictionary of IEEE Standards Terms* [B1] should be referenced for terms not defined in this clause. This clause defines terms used in the SIMICA set of standards. A clear understanding of the following terms with respect to testability and diagnosability is particularly important in order to understand this standard.

**3.1 architectural device:** In the context of software systems, a means that facilitates the arrangement of software components or model elements to accomplish a specific purpose.

**3.2 conceptual information model:** An information model that is independent of any particular instantiation form, i.e., is never intended to be realized.

**3.3 concrete information model:** An information model that is specialized to take account of a particular instantiation method or data exchange format.

**3.4 diagnostic maturation:** The process of monitoring diagnostic system predicted vs. actual performance to identify and implement corrective action. The goal is to enhance diagnostic effectiveness throughout the product life cycle. Diagnostic elements that may benefit from the maturation process include (but are not limited to) diagnostic models, system performance models, test programs, and even product design improvements.

**3.5 information model:** A formal description of types of ideas, facts, and processes that together form a model of a portion of interest of the real world and which provides an explicit set of interpretation rules.

**3.6 level of indenture:** A hierarchical partition in a physical or functional system decomposition.

<sup>3</sup> IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (<http://standards.ieee.org>).

<sup>4</sup> The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

<sup>5</sup> This IEEE standards project was not approved by the IEEE-SA Standards Board at the time this publication went to press. For information about obtaining a draft, contact the IEEE.

<sup>6</sup> ISO publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembe, CH-1211, Geneva 20, Switzerland/Suisse (<http://www.iso.ch>). ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th floor, New York, NY 10036, USA (<http://www.ansi.org/>).

**3.7 system:** A) A collection of entities to be processed by applying a top-down, hierarchical approach. B) A collection of elements forming a collective, functioning entity. C) A collection of hardware or software components necessary for performing a function.

## 4. Backgrounds

### 4.1 Diagnostic maturation

The process of maturing a system diagnostic design begins in the system's conceptual design phase and continues throughout the system life cycle. Generally diagnostic design are demonstrated prior to delivery of the first unit to meet the diagnostic and health management requirements that have been levied. However, once a system is fielded and used in an operational environment, unexpected and unplanned system level design interactions, operational and environmental stresses, and other influences tend to reveal deficiencies in the diagnostic capabilities. When such deficiencies result in a weapon system readiness or cost of ownership problem, remedial actions must be taken:

- The performance issue must be clearly characterized in terms of scope, impact, and ownership.
- Root cause analysis must be performed to identify potential courses of actions, including adjustments, improvements, or refinements to support and operational elements.
- The resulting courses of action must be deployed in a manner that is both economically feasible and consistent with product functional and operational requirements.

The diagnostic maturation process requires access to design, maintenance, and logistics support information sources. Aggregation of system failure and performance data from disparate information sources is a multidisciplinary issue, i.e., relevant data exists within the purview of engineering, product support, and other organizations. Each of these organizations is also multifaceted in nature. Engineering organizations, for example, are composed of sub-disciplines such as design, reliability and maintainability, etc. Compounding the problem is the fact that the data of interest resides in multiple systems each with different owners where it does exist—and it should be recognized that some data that is desirable to have might not be captured in data systems at all.

The problem can be reduced by operating with a single, unified conceptual view of the data to be extracted from multiple sources. Therefore, the challenge is how to extract the information of interest—heterogeneous system failure and performance data—from the disparate data systems where it resides. The idea is to identify the information of interest—and in fact, to create an information model so that it is clear what that data is and what it really means in the context of system usage and analysis—then determine the process for obtaining that information. Once data requirements and availability are determined, one can then begin trying to solve the puzzle of how to integrate the heterogeneous information to be obtained, i.e., how to relate the vital information from any one system to the others that capture related information but using disparate data formats and semantics.

Data collection is a prerequisite for the maturation process, and the product data that is typically required for maturation analysis is generally stored in heterogeneous data systems. This standard will facilitate improved access, retrieval, and integration of the requisite information from these data systems. The intent of the SIMICA family of specifications is to provide an implementation-independent specification for a software interface to information systems containing data pertinent to the diagnosis and maintenance of complex systems consisting of hardware, software, or any combination thereof. These interfaces will support the creation of service definitions for application programming interfaces (API) for the access, exchange, and analysis of historical diagnostic and maintenance information. The use of formal information models will facilitate exchanging historical maintenance information between information systems and analysis tools, supporting the creation of open system software architectures for maturing system

diagnostics. The component standards will further enhance the exchange of instance information between conforming applications through the use of standard exchange formats.

More information on the diagnostic maturation process and associated issues can be found in *The Authoritative Dictionary of IEEE Standards Terms* [B1], Wilmering 2001 [B5], and Wilmering, Yuan, and VanRossum 2003 [B6].

## 4.2 Relationship to IEEE Std 1232 (AI-ESTATE)

IEEE Std 1232 is an information exchange standard for test and diagnosis. The standard defines the architecture of an AI-ESTATE-conformant system, information models for knowledge and data exchange, and a set of standard software services to be provided by a diagnostic reasoner in an open-architecture test environment. Because it provides formal definitions (via information modeling) of the information required for test and diagnosis, it will serve as a basis for information requirements in this standard. The SIMICA information model directly references the IEEE Std 1232's Dynamic Context Model historical records of diagnostic sessions. This facilitates relating diagnostic session information to associated maintenance actions (as specified in IEEE P1636.2/D3.0). Such associations support analysis and maturation of diagnostic models and reasoner methods.

## 4.3 Relationship to IEEE 1636 (SIMICA) component standards

As described in Figure 1 and accompanying text, the SIMICA family of standards is composed of this specification, IEEE Std 1636.1 Test Results and Session Information (TRSI), and IEEE P1636.2 Maintenance Action Information (MAI). The latter two documents specify concrete information models supporting software interfaces for SIMICA data elements and are related through the conceptual information model provided in this standard.

IEEE Std 1636.1 provides a software interface between components of an automatic test system (e.g., between test executive and diagnostic reasoner) where test results need to be shared. It will therefore facilitate the capture of test results data in storage devices and databases, supporting online and offline analysis of system test performance.

IEEE P1636.2 provides a software interface between components of a test system and applications in a maintenance environment where maintenance action information needs to be shared. It will facilitate the capture and exchange of Unit Under Test-specific maintenance information, supporting online and offline analysis of the maintenance process.

It is anticipated that there will be additional SIMICA component standards added in the future.

## 5. Information model

This clause contains the specification for the SIMICA information model. The model is defined using EXPRESS (ISO 10303-11:1994).

(\* EXPRESS Specification starts here \*)

(\*

## 5.1 SIMICA\_MODEL

The EXPRESS schema "SIMICA\_MODEL" defines a high-level information model representing the essential information (and associated semantics) required to support system maintenance activities. The intent of this schema is to provide a conceptual model of maintenance information relative to system diagnostic maturation.

EXPRESS specification:

\*)

SCHEMA SIMICA\_MODEL;

USE FROM AI\_ESTATE\_COMMON\_ELEMENT\_MODEL  
(DiagnosticModel AS AIESTATEModel);

USE FROM AI\_ESTATE\_DYNAMIC\_CONTEXT\_MODEL  
(ActualRepairItem,  
Session);

USE FROM SIMICA\_MAI\_MODEL  
(MaintenanceActionInformation);

USE FROM TEST\_RESULTS\_MODEL  
(TestResults);

(\*

### 5.1.1 AvailabilityModel

Entity "AvailabilityModel" is a subtype of "DesignModel" and shall represent a model of the ability of the system to perform without failure (reliability) and to be restored to service quickly (a function of both maintainability and the level and accessibility of support resources). Availability is expressed as a steady-state probability that a component or system is in service. Specific details for this model will be defined in a separate component of the SIMICA family of standards.

EXPRESS specification:

\*)

ENTITY AvailabilityModel  
  
SUBTYPE OF (DesignModel);  
  
END\_ENTITY;

(\*

### 5.1.2 CostModel

Entity "CostModel" is a subtype of "DesignModel" and shall represent a model of the costs associated with the manufacture, operation, and support of a system.

EXPRESS specification:

```
*)  
  
    ENTITY CostModel  
  
        SUBTYPE OF (DesignModel);  
  
    END_ENTITY;  
  
(*
```

### 5.1.3 DesignLibrary

Entity "DesignLibrary" shall represent a library of design documents for a particular system. As a supertype, DesignLibrary is a container of documentation relevant to the system design.

EXPRESS specification:

```
*)  
  
    ENTITY DesignLibrary;  
  
    END_ENTITY;  
  
(*
```

### 5.1.4 DesignModel

Entity "DesignModel" shall provide a container for a set of system models characterizing different aspects of a system design.

EXPRESS specification:

```
*)  
  
    ENTITY DesignModel;  
  
    END_ENTITY;  
  
(*
```

### 5.1.5 DesignSpecification

Entity "DesignSpecification" is a subtype of "DesignLibrary" and shall represent the requirements and specification documentation for a particular system.

EXPRESS specification:

```
*)  
  
ENTITY DesignSpecification  
    SUBTYPE OF (DesignLibrary);  
  
END_ENTITY;  
  
(*
```

### 5.1.6 DiagnosticModel

Entity "DiagnosticModel" is a subtype of "DesignModel" and shall represent a model of a system to be used in determining the cause of a particular behavior or set of observations about the system. Typically, a diagnostic model is used to determine the cause of failure in a system in order to initiate a repair.

EXPRESS specification:

```
*)  
  
ENTITY DiagnosticModel  
    SUBTYPE OF (DesignModel);  
  
    modelInstance : OPTIONAL SET [1:?] OF AIESTATEModel;  
  
END_ENTITY;  
  
(*
```

Attribute definitions:

modelInstance : Attribute "modelInstance" shall identify a set of AI-ESTATE-conformant diagnostic models associated with the system of interest.

### 5.1.7 DrawingPackage

Entity "DrawingPackage" is a subtype of "DesignLibrary" and shall represent the elements necessary for the specification of system drawings.

EXPRESS specification:

```
*)  
  
ENTITY DrawingPackage  
    SUBTYPE OF (DesignLibrary);  
  
END_ENTITY;  
  
(*
```

### 5.1.8 MaintainabilityModel

Entity "MaintainabilityModel" is a subtype of "DesignModel" and shall represent a model of characteristics defining the servicing and preventive and corrective maintenance to be conducted on a repair item.

EXPRESS specification:

```
*)  
  
ENTITY MaintainabilityModel  
    SUBTYPE OF (DesignModel);  
  
END_ENTITY;  
  
(*
```

### 5.1.9 MeasurementAsset

Entity "MeasurementAsset" is a subtype of "TestResource" and shall represent the hardware or software component used to monitor and evaluate the state, condition, or characteristics of a System Under Test.

EXPRESS specification:

```
*)  
  
ENTITY MeasurementAsset  
    SUBTYPE OF (TestResource);  
  
END_ENTITY;  
  
(*
```

### 5.1.10 OperationalContext

Entity "OperationalContext" shall represent the conditions under which the system was operating at the time test or performance information was captured.

EXPRESS specification:

```
*)  
  
ENTITY OperationalContext;  
  
END_ENTITY;  
  
(*
```

### 5.1.11 ProductionSpecification

Entity "ProductionSpecification" is a subtype of "DesignLibrary" and shall represent requirements and specification documentation for the manufacturing and production of a particular system.

EXPRESS specification:

\*)

```
ENTITY ProductionSpecification  
    SUBTYPE OF (DesignLibrary);  
END_ENTITY;
```

(\*

### 5.1.12 ReliabilityModel

Entity "ReliabilityModel" is a subtype of "DesignModel" and shall represent a model of the reliability (i.e., the conditional probability of failure given some amount of operating time) of a system or portions of the system.

EXPRESS specification:

\*)

```
ENTITY ReliabilityModel  
    SUBTYPE OF (DesignModel);  
END_ENTITY;
```

(\*

### 5.1.13 SafetyModel

Entity "SafetyModel" is a subtype of "DesignModel" and shall represent a model of the safety (i.e., the conditional probability of operating in a non-hazardous state given some amount of operating time) of a system or portions of the system.

EXPRESS specification:

\*)

```
ENTITY SafetyModel  
    SUBTYPE OF (DesignModel);  
END_ENTITY;
```

(\*

#### 5.1.14 SupplyChainModel

Entity "SupplyChainModel" is a subtype of "DesignModel" and shall represent a model of the processes of supply, manufacture, and distribution.

EXPRESS specification:

\*)

```
ENTITY SupplyChainModel
    SUBTYPE OF (DesignModel);
END_ENTITY;
```

(\*

#### 5.1.15 SupportAsset

Entity "SupportAsset" is a subtype of "TestResource" and shall represent the hardware or software required to implement the test.

EXPRESS specification:

\*)

```
ENTITY SupportAsset
    SUBTYPE OF (TestResource);
END_ENTITY;
```

(\*

#### 5.1.16 SystemCollection

A SystemCollection shall consist of any aggregation of complex systems, for example a set of airplanes collected into a fleet or a set of local area networks collected into a wide area network (WAN).

EXPRESS specification:

\*)

```
ENTITY SystemCollection;
    element : SET [1:?] OF SystemUnderTest;
END_ENTITY;
```

(\*

Attribute definitions:

element : Attribute "element" shall identify the set of systems making up a SystemCollection.

### 5.1.17 SystemDesign

Entity "SystemDesign" shall represent the specification and design of a particular product or system. SystemDesign shall consist of a set of models, each representing the characteristics of different aspects of the system's construction, function, cost, support, etc. Additional documentation is also required for a SystemDesign, including drawings, requirements, specifications, manuals, etc.

EXPRESS specification:

\*)

```
ENTITY SystemDesign;
```

```
documentation : SET [1:?] OF DesignLibrary;
```

```
systemModel : SET [1:?] OF DesignModel;
```

```
END_ENTITY;
```

(\*

Attribute definitions:

documentation : Attribute "documentation" shall provide a set of design documents supporting the system.

systemModel : Attribute "systemModel" shall identify a set of design models for the system.

### 5.1.18 SystemUnderTest

Entity "SystemUnderTest" shall represent the relevant product-level attributes of a system undergoing test and diagnosis. It is also the collector for all information relevant to the System Under Test.

EXPRESS specification:

\*)

```

ENTITY SystemUnderTest;

    component                : OPTIONAL SET [1:?] OF
                             SystemUnderTest;

    utilizedResource         : SET [1:?] OF TestResource;

    test                     : SET [1:?] OF TestProcedure;

    design                   : SystemDesign;

    environment              : OperationalContext;

    testConfiguration        : TestConfigurationSpecification;

    diagnosedItem            : SET [1:?] OF ActualRepairItem;

    maintenancePerformed     : SET [1:?] OF
                             MaintenanceActionInformation;

    recordedResult           : SET [1:?] OF TestResults;

    recordedDiagnostics      : SET [1:?] OF Session;

INVERSE

    parent                   : SET [1:?] OF SystemUnderTest FOR
                             component;

END_ENTITY;

```

(\*)

Attribute definitions:

component	: Attribute "component" shall identify the set of components or subunits at the next level of indenture for the current system under test.
utilizedResource	: Attribute "utilizedResource" shall identify the set of test resources that has been used to test and support a specific system under test.
test	: Attribute "test" shall identify the set of test procedures that can be run on the system under test.
design	: Attribute "design" shall identify all of the design documentation, including documents and models, supporting the system under test.
environment	: Attribute "environment" shall identify the environmental conditions

under which the system under test operates.

- testConfiguration : Attribute "testConfiguration" shall identify the specific configuration of the test resources used to support the system.
- diagnosedItem : Attribute "diagnosedItem" shall identify the actual repair items being diagnosed that generated the diagnostic history of the SystemUnderTest (as referenced in the IEEE Std 1232 Dynamic Context Model).
- maintenancePerformed : Attribute "maintenancePerformed" shall identify a set of actual maintenance actions taken on the system as specified in IEEE P1636.2/D3.0.
- recordedResult : Attribute "recordedResult" shall provide the set of test results that have been recorded in a test session for the system (as specified by the Test Results EXPRESS schema in IEEE Std 1636.1-2007). These results must correspond to the tests used in a diagnostic session (as recorded in the AI-ESTATE EXPRESS schema).
- recordedDiagnostics : Attribute "recordedDiagnostics" shall provide the set of diagnostic results that have been recorded in a session for the system (as specified in the AI-ESTATE EXPRESS Dynamic Context Model schema).
- parent : Inverse attribute "parent" shall identify the next higher assembly of the system to which a particular component or subunit belongs.

#### 5.1.19 TestConfigurationSpecification

Entity "TestConfigurationSpecification" is a subtype of "TestSpecification" and shall represent the resources needed to describe the resources required to test the system. This includes (but is not limited to) information about test instruments, test adapters, test software, and ancillary equipment. A potential interface to this information as it applies to Automatic Test Equipment is specified in IEEE Std 1671.4 Test Configuration Information [B3].

EXPRESS specification:

\*)

```
ENTITY TestConfigurationSpecification
    SUBTYPE OF (TestSpecification);

END_ENTITY;
```

(\*

### 5.1.20 TestDescription

Entity "TestDescription" is a subtype of "TestSpecification" and shall represent the collection of test requirements, setup procedures, stimuli, measurements, and anticipated results. A potential interface to this information as it applies to Automatic Test Equipment is specified in IEEE Std 1671.1 Test Description Information [B2].

EXPRESS specification:

\*)

```
ENTITY TestDescription
    SUBTYPE OF (TestSpecification);
END_ENTITY;
```

(\*

### 5.1.21 TestProcedure

Entity "TestProcedure" shall represent the process by which a test is to be performed. This process includes identification of required test resources, the setup process, stimuli to be applied, measurements to be taken, and specifications for mapping measurements into test outcomes.

EXPRESS specification:

\*)

```
ENTITY TestProcedure;
    recordedResult      : SET [1:?] OF TestResults;
    requirement         : TestDescription;
INVERSE
    targetResource     : TestResource FOR process;
END_ENTITY;
```

(\*

Attribute definitions:

- recordedResult : Attribute "recordedResult" shall provide the set of test results that have been recorded in a test session for the system (as specified by the Test Results EXPRESS schema in IEEE Std 1636.1-2007). These results must correspond to the tests used in a diagnostic session (as recorded in the AI-ESTATE EXPRESS schema).
- requirement : Attribute "requirement" shall identify the test description document that outlines the specific requirements to be satisfied by the test procedure.
- targetResource : Inverse attribute "targetResource" shall identify a specific test resource for which the test procedure was developed.

**5.1.22 TestResource**

Entity "TestResource" shall represent the class of equipment, tools, monitors, etc., that can be used to obtain test and performance information about a system under test.

EXPRESS specification:

\*)

```
ENTITY TestResource;
```

```
    process : SET [1:?] OF TestProcedure;
```

```
END_ENTITY;
```

(\*

Attribute definitions:

- process : Attribute "process" shall identify a set of test procedures that can be performed using a specific test resource class (such as a specific model of voltmeter, or a certain kind of torque wrench, etc.).

### 5.1.23 TestSpecification

Entity "TestSpecification" is a subtype of "DesignLibrary" and shall represent requirements and specification documentation for the test, evaluation, acceptance, and maintenance of a particular system.

EXPRESS specification:

\*)

```
ENTITY TestSpecification
    SUBTYPE OF (DesignLibrary);
END_ENTITY;
```

(\*

### 5.1.24 TestabilityModel

Entity "TestabilityModel" is a subtype of "DesignModel" and shall represent a model of the controllability, observability, accessibility, and diagnosability of a system.

EXPRESS specification:

\*)

```
ENTITY TestabilityModel
    SUBTYPE OF (DesignModel);
END_ENTITY;
END_SCHEMA;
```

(\*

IECNORM.COM : Click to view the full PDF of IEC 61636:2016

5.1.25 EXPRESS-G Diagrams

Figure 2 and Figure 3 provide the graphical representation of the SIMICA EXPRESS information model.

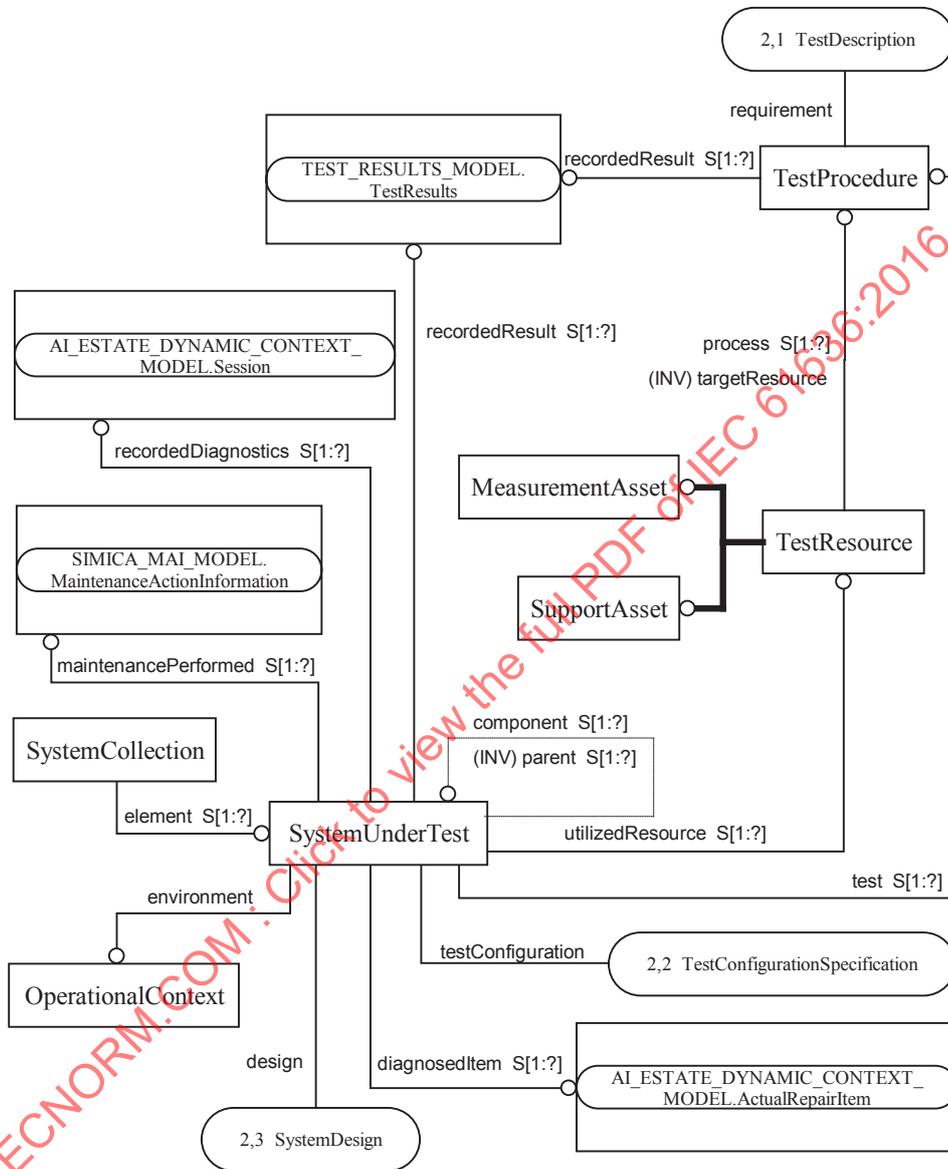


Figure 2—SIMICA\_MODEL EXPRESS-G diagram 1 of 2

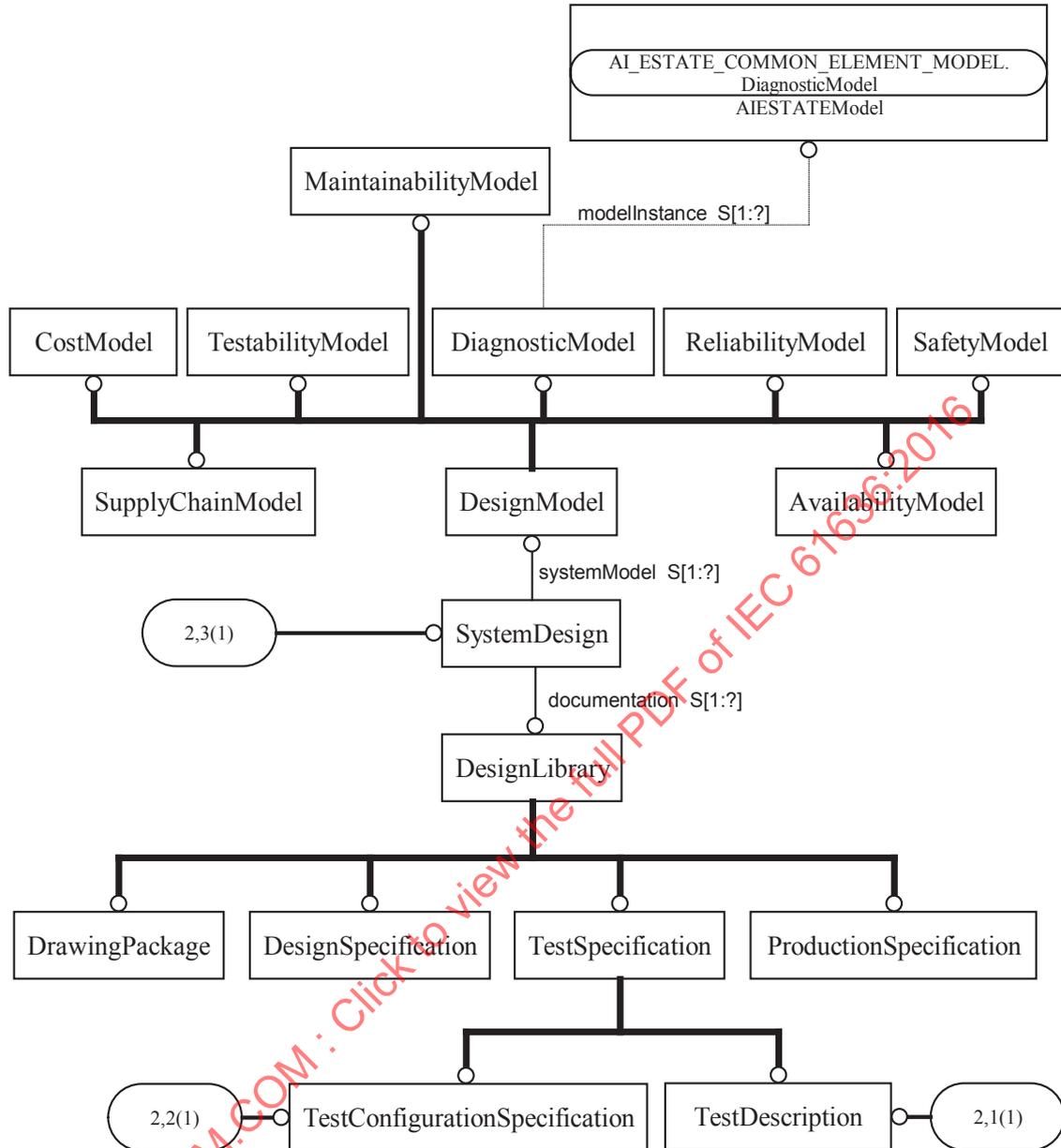


Figure 3—SIMICA\_MODEL EXPRESS-G diagram 2 of 2

## 6. Conformance

This standard specifies a unifying top-level conceptual information model to provide context for the SIMICA model family of specific software interfaces as shown in Figure 1. The elements of this top-level model are not intended to be instantiated at the data level—that is the purpose of the component models as shown in Figure 1. The conceptual information model contained in this standard identifies information components to be specified by other component specifications in the SIMICA family. Conformance requirements to those components can be found in their respective standards.

## Annex A

(informative)

### Bibliography

[B1] IEEE 100™, *The Authoritative Dictionary of IEEE Standards Terms*, 2d ed. New York, Institute of Electrical and Electronics Engineers, Inc. 2000.<sup>7, 8</sup>

[B2] IEEE P1671.1™/D4, Draft Trial-Use Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML, Exchanging Test Descriptions.<sup>9</sup>

[B3] IEEE P1671.4™/D6, Draft Trial-Use Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML, Exchanging Test Configuration Information.<sup>9</sup>

[B4] Sheppard, John W. and Wilmering, Timothy J., “Recent Advances in IEEE Standards for Diagnosis and Diagnostic Maturation,” *Proceedings IEEE Aerospace Conference*, Big Sky, MT, 2006.

[B5] Wilmering, Timothy J., “Semantic Requirements on Information Integration for Diagnostic Maturation,” *Autotestcon 2001 Proceedings*, Anaheim, CA, 2001.

[B6] Wilmering, Timothy J., Yuan, Jun, and VanRossum, David, “A Metadata Architecture for Mediated Integration of Product Usage Data,” *Proceedings of Autotestcon 2003*, Anaheim, CA, 2003.

---

<sup>7</sup> The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

<sup>8</sup> IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (<http://standards.ieee.org>).

<sup>9</sup> This IEEE standards project was not approved by the IEEE-SA Standards Board at the time this publication went to press. For information about obtaining a draft, contact the IEEE.

## Annex B

(informative)

### Overview of EXPRESS

The models defined in this standard use the ISO 10303-11 EXPRESS language and the supporting EXPRESS-G graphical notation for their specification. To quote from ISO 10303-11, “*EXPRESS* is the name of the formal information modeling language used to specify the information requirements of other parts of this International Standard... The language focuses on the definition of *entities*, which are the things of interest. The definition of entities is in terms of data and behavior. Data represents the properties by which an entity is realized and behavior is represented by constraints.”

Within EXPRESS, models are defined using a simple hierarchy partitioned along *schemata*, *entities*, and *attributes*. Further, legal values of attributes are defined through *constraints* on those attributes. The scope of the language is to define the information to be used or generated by a system or process and is not intended to define database formats, file formats, or exchange formats. Further, EXPRESS is not intended to be used as a programming language since it contains no facilities for input/output, exception handling, or information processing.

In the following sections, the major elements of an EXPRESS model will be described. When available, the corresponding representation of the element in EXPRESS-G will also be provided.

#### B.1 Schema

A *schema* is defined to be “a collection of items forming part or all of a model.”

The syntax of a schema definition consists of

```
SCHEMA schema_id ';' schema_body END_SCHEMA ';' 
```

and a schema is represented in EXPRESS-G as

schema_id

#### B.2 Entity

An *entity* is defined to be “a type which represents information for processing purposes, based on explicit or implicit agreements about the meaning of the data.” Within each schema of SIMICA, the primary data types are defined as entities, many of which have identifiers as indicated by “description” in the entity definition.

The syntax of an entity definition consists of

```
ENTITY entity_id [supertype] [subtype] ';' entity_body END_ENTITY  
' ; '
```

and an entity is represented in EXPRESS-G as



### B.3 Attribute

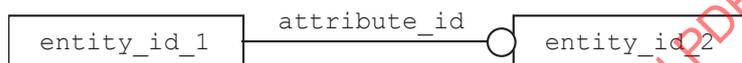
An *attribute* is defined to be “a trait, quality or property that is a characteristic of an entity.” Attributes provide the primary elements of the definition of the entity body. Since entities are type definitions, frequently, attributes are of types as defined by other entities.

Attributes may be optional or required. Frequently, attributes in SIMICA are defined to be sets, and many of these sets have minimum cardinality of zero. An attribute that is optional is intended to be different from an attribute with cardinality of zero. For an optional attribute, an instantiated model may or may not include that attribute. If an attribute is required but may have a cardinality of zero, then a placeholder for that attribute shall be included in the instantiation even though no value is assigned.

The syntax of a simple attribute definition consists of

```
attribute_id ':' [OPTIONAL] base_type ';' 
```

Required attributes are defined between entities or an entity and a type and are represented in EXPRESS-G as



In this case, `entity_id_1` has attribute `attribute_id` which has type `entity_id_2`. The circle on the line can be treated as an “arrow-head” which determines the direction of the relationship between the two entities.

Similarly, optional attributes are defined between entities or an entity and a type and are represented in EXPRESS-G as



Finally, attributes can be defined to have an inverse relationship in which the named attribute (e.g., `entity_id_2` above) is associated with the declared entity (e.g., `entity_id_1` above). Within the EXPRESS specification, an inverse relationship is identified with the INVERSE keyword, and attributes following INVERSE define the inverse attribute.

The syntax for an inverse attribute consists of

```
attribute_id ':' [ SET [ '[' bound_1 ':' bound_2 ']' ] OF ]  
entity_id FOR attribute_id ';' 
```

An example illustrating the use of inverse attributes can be represented in EXPRESS-G as

