IEC 61636-99

Edition 1.0 2016-11

INTERNATIONAL STANDARD

IEEE Std 1636.99™

Software Interface for Maintenance Information Collection and Analysis (SIMICA): Common Information Elements

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 25.040.01; 35.060

ISBN 978-2-8322-3685-7

Contents

SOFTWARE INTERFACE FOR MAINTENANCE INFORMATION COLLECTION AND ANALYSIS (SIMICA): COMMON INFORMATION ELEMENTS

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 61636-99/IEEE Std 1636.99 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
1636.99 (2013)	91/1361/FDIS	91/1372/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.

IEEE Standard for Software Interface for Maintenance Information Collection and Analysis (SIMICA): Common Information Elements

Sponsor

IEEE Standards Coordinating Committees on Test and Diagnosis for Electronic Systems (SCC20)

Approved 23 August 2013

IEEE-SA Standards Board

Abstract: This standard is intended to promote and facilitate interoperability between components of SIMICA. The standard defines EXPRESS information models and XML schemas that together define the common information elements supporting these interfaces.

Keywords: automated test system (ATS), eXtensible markup language (XML), IEEE 1636.99™, session information, Software Interface for Maintenance Information Collection and Analysis (SIMICA), test results, XML schema

IEEE Introduction

This introduction is not part of IEEE Std 1636.99™-2013, IEEE Standard for Software Interface for Maintenance Information Collection and Analysis (SIMICA): Common Information Elements.

Maintainers of complex systems require the ability to capture and share historical test and maintenance-related information in a way that supports such activities as performance analysis, post-production product improvement, maintenance process improvement, and diagnostic maturation. Principal stakeholders of this project include but are not limited to maintenance organizations within various Departments/Ministries of Defense, the commercial airlines, the automotive industry, and the telecommunications industry. This standard is being developed as a component of the IEEE 1636™ Software Interface for Maintenance Information Collection and Analysis (SIMICA) project. SIMICA's purpose is to specify a software interface for access, exchange, and analysis of product diagnostic and maintenance information. Maintenance action information provides a subset of the data needed to satisfy SIMICIA requirements.

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.

The XML schema described in this standard where appropriate utilizes and references components of the IEEE Std 1671™ schema set.

It is anticipated that these schemas will be used throughout industries that utilize diagnostic and maintenance data as an exchange format that can be understood by humans or machines. In order to ensure wide acceptance throughout the user community, the schemas have been designed to encompass a broad range of use cases. To accommodate use cases beyond the released design, the schemas provide means for user extensibility.

Software Interface for Maintenance Information Collection and Analysis (SIMICA): Common Information Elements

1. Overview

1.1 General

This standard, which is a component of the Software Interface for Maintenance Information Collection and Analysis (SIMICA) standard, was developed by the Diagnostic and Maintenance Control Subcommittee of the IEEE Standards Coordinating Committee 20 (SCC20) on Test and Diagnosis for Electronic Systems to provide standard, unambiguous definitions of common SIMICA element semantics, and interrelationships.

This standard specifically describes a set of formal specifications consisting of the logical representation of the information that is common between IEEE Std 1636.1™ and IEEE Std 1636.2™, which may be used during related diagnostic and maintenance processes. The information model contained in this document provides a normative formal specification of the information concepts and precise semantics that support the unambiguous exchange of information between producers and consumers in a platform-independent manner.

The schemas described in this document are intended to be shared by all SIMICA "dot" standards. The Express schema in this standard is based on ISO 10303-11:1994 [B9][1]. The XML schema associated with this standard is based on the W3C eXtensible Markup Language (XML) 1.0 Recommendation [B1][2].

1.2 Application of this documents annexes

This document includes three annexes. Of these three, two are normative (Annex A and Annex B).

Annex A contains the description of each of the XML schema elements and types.

Annex B contains the description of the EXPRESS and EXPRESS-G model elements.

Annex C is informative, and thus is provided strictly as information, for both users and maintainers of this document.

1.3 Scope

The SIMICA family of standards provides an implementation-independent software interfaces to information systems containing data pertinent to the diagnosis and maintenance of complex systems consisting of hardware, software, or any combination thereof. This standard defines EXPRESS information models and XML schemas that together define the common information elements supporting these interfaces.

1.4 Referenced IEEE Standards

SIMICA Common makes reference to IEEE Std 1671™-2010 Annex B.1. This normatively referenced IEEE standard, when utilized, is therefore considered part of the SIMICA definition.

1.5 Application

This standard provides a specification for information shared by SIMICA "dot" standards (e.g., IEEE Std 1636.1, IEEE Std 1636.2). Anticipated users of this standard include the following:

a) System developers

b) System maintainers

c) Reliability, maintainability, and diagnostic analytical applications

1.6 Conventions used in this document

1.6.1 General

In accordance with *IEEE Standards Style Manual* [B3], any schema examples will be shown in `Courier` font. In cases where instance document examples are necessary to depict clearly use of a schema type or element, such examples will also be shown in `Courier` font. When the characters "..." appear in an example, it indicates that the example component is incomplete.

[1] The numbers in brackets correspond to those of the bibliography in Annex C.
[2] W3C is a registered trademark of the World Wide Web Consortium.

All simple types, complex types, attribute groups, and elements will be listed; explanatory information will be provided, along with examples if additional clarification is needed. The explanatory information shall include information on the intended use of the elements and/or attributes where the name of the entity does not clearly indicate its intended use. For elements derived from another source type (e.g., an abstract type), only attributes which extend the source type shall be listed; details regarding the base type shall be listed along with the base type.

When referring to an attribute of an XML element, the convention of *[element]@[attribute]* shall be used. In cases where an attribute name is referred to with no associated element, the attribute name shall be enclosed in single quotes.

In tables that describe XML elements, the column "Use" indicates the occurrence constraints for each element.

a) "Required" indicates that the element shall appear exactly once.

b) "Optional" indicates that the element may appear once or not at all.

c) "1..∞" indicates that the element shall appear at least once and may appear multiple times.

d) "0..∞" indicates that the element may appear multiple times, once, or not at all.

All specifications for the EXPRESS language are given in the `Courier` type font which includes references to entity and attribute names in the supporting text.

This standard uses the vocabulary and definitions of relevant IEEE standards. In case of conflict of definitions, except for those portions quoted from standards, the following precedence shall be observed: 1) Clause 3, and 2) The *IEEE Standards Dictionary Online* [B2][3].

1.6.2 Word usage

In accordance with *the IEEE Standards Style Manual* [B3], the word *shall* is used to indicate mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*). The use of the word *must* is used only to describe unavoidable situations. The use of the word *will* is only used in statements of fact.

The word *should* is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability (*can* equals *is able to*).

[3] *IEEE Standards Dictionary Online* subscription is available at:
http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html

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.

IEEE Std 1636™-2009, IEEE Standard for Software Interface for Maintenance Information Collection and Analysis (SIMICA).[4, 5]

IEEE Std 1671™-2010, IEEE Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML.

[4] IEEE publications are available from The Institute of Electrical and Electronics Engineers (http://standards.ieee.org/).
[5] The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

maintenance: Activity intended to keep equipment (hardware) or programs (software) in satisfactory working condition, including replacements, adjustments, repairs, software/firmware updates, and program improvements. Maintenance can be preventative or corrective (adapted from MIL-STD-1309D [B12]).

particle (in eXtensible Markup Language (XML) schema): A kind of component.

sequence (in XML schema): A compositor for model group schema components which specifies that subordinate elements in an instance document must correspond, in order, to the specified particles.

3.2 Acronyms and abbreviations

AI-ESTATE	Artificial Intelligence Exchange and Service Tie to All Test Environments
ATML	Automatic Test Markup Language
ATS	automatic test system
ISO	International Organization for Standardization
OID	object identifier
SIMICA	Software Interface for Maintenance Information Collection and Analysis
URL	uniform resource locator
URN	uniform resource name
UUT	unit under test
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

4. SIMICA common elements

4.1 SIMICA common element partitioning

Common elements provide for the definition of XML types and attributes that are utilized within more than one SIMICA family XML schema.

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

SIMICA Common elements as a result are simply a toolbox for the SIMICA family XML schemas to include.

SIMICA defines one common element XML schema:

SIMICACommon.xsd: provides SIMICA-unique types and attributes.

This SIMICA common element XML schema is defined in Annex A.

The EXPRESS and EXPRESS-G models of SIMICA Common elements are defined in Annex B.

Every SIMICA family component XML schema shall include the SIMICA common element XML schema.

Every SIMICA family component shall reference the SIMICA common element EXPRESS models.

4.2 Use of the IEEE Std 1671 Common.xsd schema

The SIMICACommon.xsd schema includes the IEEE Std 1671 Common.xsd schema.

Having the SIMICA common elements XML schema include the ATML common elements XML schema allows for a consistent definition of shared XML types and prevents each XML schema from defining XML types used by other SIMICA and ATML family XML schemas.

5. EXPRESS model, EXPRESS-G diagram, and XML schema names and locations

The IEEE provides a download site for material published in association with IEEE Standards, presented in machine friendly format. This material is digital rights management restricted use material. The SIMICA family of standards utilizes this download site to allow easy accessibility to all of the SIMICA family EXPRESS models and XML schemas (and in some cases, example XML instance documents). As depicted by Figure 1, the IEEE download site (http://standards.ieee.org/downloads/) contains several folders, each folder labeled by an associated IEEE standards number (e.g., IEEE 1636 standards are in the 1636 folder). Each folder under the “base” IEEE standards number contains the material (XML schemas, etc) for that family member. Family members are identified by their “dot” standard number (if it is a “dot” standard) 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 1 depicts a portion of the IEEE download site, as it pertains to the SIMICA family of standards.

http://standards.ieee.org/downloads

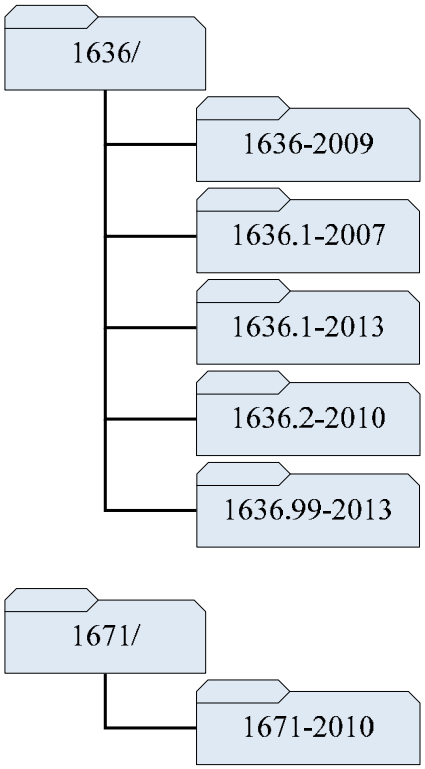

Figure 1—SIMICA-related IEEE download site structure

The IEEE Std 1636.99–associated XML schema's name, and the IEEE download site folder locations; where the XML schema shall be located, is as defined in Table 1. Where the IEEE 1636.99 associated EXPRESS model shall be located, is as defined in Table 2.

Table 1—IEEE Std 1636.99 XML schema names and folder locations

Component	Defined in	XML schema name	IEEE download site folder (see Figure 1)
SIMICA Common	Annex A	SIMICACommon.xsd	1636/1636.99-2013

Table 2—IEEE Std1636.99 EXPRESS model and diagram names and folder locations

Component	Defined in	EXPRESS model name	IEEE download site folder (see Figure 1)
SIMICA Common EXPRESSS Model	Annex B.1	1636.99.exp	1636/1636.99-2013

The XML schema identified in Table 1 includes the ATML common XML schema. The ATML common element XML schema name and the IEEE download site folder locations and where the XML schema shall be located is as defined in Table 3.

Table 3—ATML common element XML schema names and locations

Component	Defined in	XML schema name	IEEE download site folder
Common	IEEE Std 1671-2010 Annex B.1	Common.xsd	1671/1671-2010

6. Conformance

The minimal expectation for XML instance documents conformant with this document shall be that a populated XML instance is considered valid if it complies with:

a) The SIMICA Common XML schema (defined in Annex A of this document, and available as described in Clause 5).

b) The SIMICA Common EXPRESS model (defined in Annex B of this document, and available as described in Clause 5).

c) The ATML Common XML schema (defined in Annex B.1 of IEEE Std 1671, and available as described in Clause 5).

7. XML schema extensibility

A provision in the XML schema of an extension mechanism is necessary to ensure the viability of the specification and allow producers and consumers of SIMICA XML instance documents to interoperate in those cases where there is a requirement to exchange relevant data that is not included in the SIMICACommon.xsd schema. 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 non-extended portions of the data as it is 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 this standard.

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 schema.

SIMICACommon.xsd supports two forms of extension:

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

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

XML schemas control the location and type of extension allowed.

An element has an extensible content model if in instance documents that element can contain elements and data beyond that specified by the schema. SIMICA schemas should explicitly identify where they can be extended. Only elements from a namespace different from the document namespace should be allowed in an extension. The schema shall use the SIMICACommon `<Extension>` type to identify where extension is allowed.

Allowing the extension of a schema using type substitution should be avoided. Schemas should mark elements defined via a simple or complex type with the block attribute set to #all if type substitution is to be avoided. Elements which use type substitution as their means of definition should set the abstract attribute to true.

Annex A

(normative)

SimicaCommon XML schema

Should the reader not have a general understanding of XML schemas, there are several XML schema tutorials available for reference. The *XML Schema Part 0: Primer* [B13], the *XML Schema Tutorial* [B14] and the *XML Schema Tutorial, Part 1* [B15] are three available on the World Wide Web. These tutorials will help with the understanding of the contents of the TestResults.xsd schema that this Annex is defining the elements of.

Prefixes utilized in this Annex are as follows:

a) The namespace prefix “c:” identifies that the type, element, or attribute is contained in the XML schema defined in Annex B.1 of IEEE Std 1671 (Common.xsd).

b) The namespace prefix “sc:” identifies that the type, element, or attribute is contained in the XML schema defined in this Annex (SIMICACommon.xsd).

A.1 SIMICACommon.xsd

A.1.1 Root element

attributeFormDefault	unqualified
elementFormDefault	qualified
targetNamespace	urn:IEEE-P1636.99:2013:SimicaCommon

A.1.2 complexType Extension

properties	**final** #all
annotation	**documentation** The Extension complex type is provided for the convenience of XML schema developers. The Extension type shall be used only as the base type of extension elements in XML schemas. Such elements are provided to permit implementers to extend a XML schema as required to meet the unique needs of their use case. Use follows the W3C standard XML extension mechanism.

A.1.3 complexType HardwareInstance

type	extension of c:HardwareInstance
properties	**base** c:HardwareInstance
children	c:DescriptionDocumentReference c:Definition c:SerialNumber c:ManufactureDate c:Calibration c:Components c:ParentComponent c:PowerOn IssueDate Warranty
annotation	**documentation** Base type for any element containing information characterizing a hardware item.

A.1.4 complexType IdentificationNumber

type	extension of c:IdentificationNumber					
properties	**base** c:IdentificationNumber					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	number	c:NonBlankString	required			**documentation** The part number of the entity.
	type	derived by: xs:string	required			**documentation** An indication of whether this is a part number, model number, or other.
	manufacturerName	c:NonBlankString	optional			**documentation** Identifies the name of the manufacturer.
	qualifier	c:NonBlankString	optional			**documentation** Provides additional descriptive data that further specifies or identifies the Identification Number.
annotation	**documentation** Base type for any element that needs to be identified.					

A.1.5 complexType MaintenanceAction

children	InstallAction RepairAction RemoveAction NoAction Description					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	actionID	xs:ID	required			**documentation** A unique identifier for the maintenance action.
	name	c:NonBlankString	required			**documentation** A user-friendly textual name for the maintenance action.
	startTime	xs:dateTime	required			**documentation** The date and time for the initiation of a maintenance event.
	endTime	xs:dateTime	optional			**documentation** The date and time for the termination of a maintenance event.
	totalTime	xs:duration	required			**documentation** The aggregate personnel time for the actual work duration of a maintenance event.
	preventative	xs:boolean	required			**documentation** Identifies whether or not the MaintenanceAction was performed as part of preventative maintenance (true) or not (false).
	actionCode	c:NonBlankString	optional			**documentation** The code that best describes the maintenance action performed. Maintenance organizations typically have a list of organization-unique codes that best describe the reasons for maintenance action performed. This attribute provides a place to record that code. The meaning and interpretation of the code is implementation-specific.
annotation	**documentation** Base type for any element defining a specific action taken in the maintenance of some item or entity.					

A.1.6 element MaintenanceAction/Description

type	c:NonBlankString
properties	**content** simple
Facets	**Kind** **Value** **Annotation** minLength 1 whiteSpace replace
annotation	**documentation** A detailed textual description of the maintenance action performed.

A.1.7 element MaintenanceAction/InstallAction

type	SystemInstance
properties	**content** complex
children	c:DescriptionDocumentReference c:Definition c:SerialNumber IssueDate Warranty
annotation	**documentation** Installation of either a repaired or new component into a unit to complete the repair.

A.1.8 element MaintenanceAction/NoAction

type	c:ItemDescription					
properties	**content** complex					
children	c:Description c:Identification c:Extension					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	version	c:NonBlankString	optional			**documentation** A string designating the version of the described item.
	name	c:NonBlankString	optional			**documentation** A descriptive or common name for the described item.
annotation	**documentation** The reason for not being able to perform maintenance on the unit.					

A.1.9 element MaintenanceAction/RemoveAction

type	SystemInstance
properties	**content** complex
children	c:DescriptionDocumentReference c:Definition c:SerialNumber IssueDate Warranty
annotation	**documentation** The removal of a component to affect a maintenance action.

A.1.10 element MaintenanceAction/RepairAction

type	RepairActionInformation
properties	**content** complex
children	ItemRepaired RepairType
annotation	**documentation** The work required to restore the unit to an operational state.

A.1.11 complexType RepairActionInformation

children	ItemRepaired RepairType
Used by	element MaintenanceAction/RepairAction
annotation	**documentation** Base type for any element defining the work required to restore the unit to an operational state.

A.1.12 element RepairActionInformation/ItemRepaired

type	SystemInstance
properties	**content** complex
children	c:DescriptionDocumentReference c:Definition c:SerialNumber IssueDate Warranty
annotation	**documentation** Identification of the item that has been repaired.

A.1.13 element RepairActionInformation/RepairType

type	RepairCode		
properties	**content** simple		
Facets	**Kind**	**Value**	**Annotation**
	enumeration	Repair	
	enumeration	Replace	
	enumeration	Reseat	
	enumeration	Alignment	
	enumeration	SoftwareUpgrade	
	enumeration	UserDefinedCode	
annotation	**documentation** Value for the types of repair actions that can be taken. Allowable values are Repair, Replace, Reseat, Alignment, SoftwareUpgrade, and UserDefinedCode.		

A.1.14 complexType SoftwareInstance

type	extension of c:SoftwareInstance
properties	**base** c:SoftwareInstance
children	c:DescriptionDocumentReference c:Definition c:SerialNumber c:ReleaseDate IssueDate Warranty
annotation	**documentation** Base type for any element that will contain information to identify a software item.

A.1.15 complexType SystemInstance

type	extension of c:ItemInstance
properties	**base** c:ItemInstance
children	c:DescriptionDocumentReference c:Definition c:SerialNumber IssueDate Warranty
Used by	**elements** MaintenanceAction/InstallAction RepairActionInformation/ItemRepaired MaintenanceAction/RemoveAction
annotation	**documentation** Base type for any element describing a specific target of maintenance against which the maintenance action has been performed.

A.1.16 complexType SystemInstanceReference

type	extension of c:ItemInstanceReference
properties	**base** c:ItemInstanceReference
children	c:InstanceDocumentReference c:Definition
annotation	**documentation** Base type for any element that require an element to reference a sc:SystemInstance which has no serial number.

A.1.17 complexType Warranty

used by	**element** SystemInstanceExtension/Warranty					
attributes	**Name**	**Type**	**Use**	**Default**	**Fixed**	**Annotation**
	warrantedBy	c:NonBlankString	required			**documentation** Name of the manufacturer which warrants the unit.
	duration	xs:duration	required			**documentation** Time the unit is covered by the warranty, e.g., one-year warranty, two-year warranty, etc.
	effectiveDate	xs:date	required			**documentation** Date when the manufacturer warranty begins to take effect.
	exclusion	c:NonBlankString	required			**documentation** Discrepancies in the unit that are not covered by the warranty.
annotation	**documentation** Base type for any element defining a manufacturer's legal responsibility for the support of a given system or item.					

A.1.18 complexType WorkOrder

type	extension of c:WorkOrder
properties	**base** c:WorkOrder
children	c:WorkOrderNumber c:WorkItemNumber c:MaintenanceLevel c:Description c:Extension
annotation	**documentation** Base type for any element defining identifying information for the item being maintained.

A.1.19 simpleType RepairCode

type	restriction of xs:string		
properties	**base** xs:string		
Used by	**element** RepairActionInformation/RepairType		
Facets	**Kind**	**Value**	**Annotation**
	enumeration	Repair	
	enumeration	Replace	
	enumeration	Reseat	
	enumeration	Alignment	
	enumeration	SoftwareUpgrade	
	enumeration	UserDefinedCode	
annotation	**documentation** Base type: restriction of xs:string Enumerations: Repair | Replace | Reseat | Alignment | SoftwareUpgrade | UserDefinedCode This shall be used as the base type for the RepairType to specify permitted UUT repair codes.		

Annex B

(normative)

SimicaCommon EXPRESS models

Should the reader not have a general understanding of EXPRESS, there are several overviews of EXPRESS available for reference. Annex B of IEEE Std 1232: [B5] and Schenk and Wilson [B11] are two available. These overviews will help with the understanding of the contents of the test results information model that this Annex is defining.

B.1 SIMICA_COMMON_MODEL_DOT_99

(*

Schema “SIMICA_COMMON_MODEL_DOT_99” provides semantic definitions of concepts common to each of the information models in the IEEE 1636 SIMICA family of standards.

*)

B.1.1 AbbreviationType

Type “Abbreviation” is a string that serves as a shortened form for a word or phrase.

EXPRESS specification:

```
*)
  TYPE AbbreviationType = STRING;
  END_TYPE;
(*
```

B.1.2 AddressComponent

Type “AddressComponent” provides a string type to be used in constructing a mailing address for an individual or an organization.

EXPRESS specification:

```
*)
  TYPE AddressComponent = STRING;
  END_TYPE;
(*
```

B.1.3 BooleanType

A data element that contains a Boolean value.

EXPRESS specification:

```
*)
  TYPE BooleanType = BOOLEAN;
  END_TYPE;
(*
```

B.1.4 ClassificationType

Defines a data element for capturing security classification marking information for a data item or document. Classification markings are user specific and can correspond to DoD markings, MoD markings, proprietary markings, etc.

EXPRESS specification:

```
*)
  TYPE ClassificationType = STRING;
  END_TYPE;
(*
```

B.1.5 Code

Type “Code” defines a string type for specifying a short-hand representation for some object, event, or occurrence. Such codes, unless otherwise specified, are assumed to be application or context specific.

EXPRESS specification:

```
*)
  TYPE Code = STRING;
  END_TYPE;
(*
```

B.1.6 Date

Type “Date” defines a string for representing date information according to the XML date type.

EXPRESS specification:

```
*)
  TYPE Date = STRING;
  END_TYPE;
(*
```

B.1.7 DateTime

Type “DateTime” defines a string for representing date and time information according to the XML dateTime type.

EXPRESS specification:

```
*)
  TYPE DateTime = STRING;
  END_TYPE;
(*
```

B.1.8 DescriptionType

Type “DescriptionType” provides a string that can be used to describe some entity.

EXPRESS specification:

```
*)
  TYPE DescriptionType = STRING;
  END_TYPE;
(*
```

B.1.9 Duration

Type “Duration” corresponds to a string representation of the XML duration data type. This data type is intended to represent a duration of time capturing Gregorian year, day, hour, minute, and second components.

EXPRESS specification:

```
*)
  TYPE Duration = STRING;
  END_TYPE;
(*
```

B.1.10 EmailAddress

Type “EmailAddress” defines a string type for specifying internet or intranet-based electronic mail addresses.

EXPRESS specification:

```
*)
  TYPE EmailAddress = STRING;
  END_TYPE;
(*
```

B.1.11 EqualityComparatorType

A string that defines the type of comparison being performed.

EXPRESS specification:

```
*)
  TYPE EqualityComparatorType = STRING;
  WHERE
    validComparators    :(SELF = 'EQ') OR
              (SELF = 'NE') OR
              (SELF = 'CIEQ') OR
              (SELF = 'CINE');
  END_TYPE;
(*
```

Formal propositions:

validComparators	Requires the comparator to be case sensitive (EQ or NE) or case insensitive (CIEQ or CINE) to cover equality and inequality.

B.1.12 Exclusion

Type “Exclusion” defines a string type for specifying legal exclusions from some contract, warranty, or other agreement.

EXPRESS specification:

```
*)
  TYPE Exclusion = STRING;
  END_TYPE;
(*
```

B.1.13 FaultID

A UUID that identifies a specific failure mode within a system item.

EXPRESS specification:

```
*)
  TYPE FaultID = UUID;
  END_TYPE;
(*
```

B.1.14 Identifier

Type “Identifier” defines a string to be used to identify a specific entity or object using some identification method other than name.

EXPRESS specification:

```
*)
  TYPE Identifier = STRING;
  END_TYPE;
(*
```

B.1.15 InequalityComparatorType

A string that defines the type of comparison being performed.

EXPRESS specification:

```
*)
  TYPE InequalityComparatorType = STRING;
  WHERE
    validComparators    :(SELF = 'GT') OR
              (SELF = 'GE') OR
              (SELF = 'LT') OR
              (SELF = 'LE')
              ;
  END_TYPE;
(*
```

Formal propositions:

validComparators Ensures the comparator used corresponds to an inequality other than NOT.

B.1.16 LogicalOperator

Type ‘LogicalOperator’ specifies a binary operator used in a logical expression.

EXPRESS specification:

```
*)
  TYPE LogicalOperator = STRING;
  WHERE
    validOperators      :(SELF = 'AND') OR
              (SELF = 'OR') OR
              (SELF = 'XOR');
  END_TYPE;
(*
```

Formal propositions:

validOperators Ensures the operator is either AND, OR, or XOR.

B.1.17 NameType

Type “NameType” defines a string to be used to name a specific entity or object.

EXPRESS specification:

```
*)
  TYPE NameType = STRING;
  END_TYPE;
(*
```

B.1.18 NonStandardUnit

A string indicating the units of a value where those units are not defined by IEEE Std 260.1.

EXPRESS specification:

```
*)
  TYPE NonStandardUnit = STRING;
  END_TYPE;
(*
```

B.1.19 NumericType

A data element that contains a floating point value.

EXPRESS specification:

```
*)
  TYPE NumericType = REAL;
  END_TYPE;
(*
```

B.1.20 QualifierType

When used, shall contain an adjective providing additional descriptive data for the ‘value’ attribute selected. For example, <...value=“Failed” qualifier=“High” .../>. In the case of <...value=“Aborted” ...> qualifier will provide essential descriptive or explanatory information regarding the reason for the subject Test not completing normally.

EXPRESS specification:

```
*)
  TYPE QualifierType = STRING;
  END_TYPE;
(*
```

B.1.21 SequenceCode

Type “SequenceCode” provides a unique identification code other than a UUID to be used in a system-specific way. The particular sequence code is application specific.

EXPRESS specification:

```
*)
  TYPE SequenceCode = STRING;
  END_TYPE;
(*
```

B.1.22 StandardUnit

A string indicating the units of a value where those units are defined by IEEE Std 260.1.

EXPRESS specification:

```
*)
  TYPE StandardUnit = STRING;
  END_TYPE;
(*
```

B.1.23 StringType

A data element that contains a string of alpha-numeric values.

EXPRESS specification:

```
*)
  TYPE StringType = STRING;
  END_TYPE;
(*
```

B.1.24 TelephoneNumber

Type “TelephoneNumber” defines a string type to be used to specify a telephone number for a specific person or organization.

EXPRESS specification:

```
*)
  TYPE TelephoneNumber = STRING;
  END_TYPE;
(*
```

B.1.25 Text

This type enables including the actual text of a document.

EXPRESS specification:

```
*)
  TYPE Text = STRING;
  END_TYPE;
(*
```

B.1.26 TypeDescription

Defines a data element to represent a categorization or classification of some object or entity based on some set of common characteristics.

EXPRESS specification:

```
*)
  TYPE TypeDescription = STRING;
  END_TYPE;
(*
```

B.1.27 URL

Type “URL” provides a string for specifying a uniform resource locator referencing a relevant document or web page.

EXPRESS specification:

```
*)
  TYPE URL = STRING;
  END_TYPE;
(*
```

B.1.28 UUID

Type “UUID” is a string that provides a “universally unique identifier” to be used to identify documents.

EXPRESS specification:

```
*)

  TYPE UUID = STRING;
  END_TYPE;
(*
```

B.1.29 Version

Type “Version” is a string used to indicate the version of an item, system, or piece of software. A string type is used to support a variety of versioning conventions.

EXPRESS specification:

```
*)
  TYPE Version = STRING;
  END_TYPE;
(*
```

B.1.30 IdentifierType

'IdentifierType' provides an enumeration indicating whether the identifier is for a part, model, or something else.

EXPRESS specification:

```
*)
  TYPE IdentifierType = ENUMERATION OF
    (PARTID,
    MODELID,
    OTHERID);
  END_TYPE;
(*
```

B.1.31 OutcomeValues

Enumerated type 'OutcomeValue' provides a list of legal outcomes for actions or tests.

EXPRESS specification:

```
*)
  TYPE OutcomeValues = ENUMERATION OF
    (ABORTED,
    BAD,
    CANDIDATE,
    DONE,
    FAILED,
    GOOD,
    NOTAVAILABLE,
    NOTKNOWN,
    NOTSTARTED,
    PASSED,
    USERDEFINED);
  END_TYPE;
(*
```

B.1.32 RepairCode

Enumerated type 'RepairCode' provides a list of legal values for the types of repair actions that can be taken.

EXPRESS specification:

```
*)
  TYPE RepairCode = ENUMERATION OF
    (REPAIRED,
    REPLACED,
    RESEATED,
    ALIGNMENT,
    SOFTWAREUPGRADE,
    USERDEFINEDCODE);
  END_TYPE;
(*
```

B.1.33 Datum

Chooses between a data type of Boolean, Numeric, String, or DateTime.

EXPRESS specification:

```
*)
  TYPE Datum = SELECT
    (BooleanType,
    NumericType,
    StringType,
    DateTime);
  END_TYPE;
(*
```

B.1.34 DesignDescriptor

Chooses between a document reference and an ItemDescription.

EXPRESS specification:

```
*)
  TYPE DesignDescriptor = SELECT
    (ItemDescription,
    DescriptionDocumentReference);
  END_TYPE;
(*
```

B.1.35 IdChoice

Chooses between a manufacturer's identification number and an end user's identification number.

EXPRESS specification:

```
*)
  TYPE IdChoice = SELECT
    (ManufacturerIdNumber,
    IdentificationNumber);
  END_TYPE;
(*
```

B.1.36 InstanceDescriptor

Chooses between a document reference and an ItemInstance.

EXPRESS specification:

```
*)
  TYPE InstanceDescriptor = SELECT
    (ItemInstance,
    DescriptionDocumentReference);
  END_TYPE;
(*
```

B.1.37 LimitType

Chooses between the type of test limit to include a mask, an expected value, a pair of limit values, or a single limit/threshold value.

EXPRESS specification:

```
*)
  TYPE LimitType = SELECT
    (Expected,
    SingleLimit,
    LimitPair,
    Mask);
  END_TYPE;
(*
```

B.1.38 SiteOrText

Chooses between a URI to locate a document or the actual text of the document itself.

EXPRESS specification:

```
*)
  TYPE SiteOrText = SELECT
    (URL,
    Text);
  END_TYPE;
(*
```

B.1.39 ActionOutcome

Defines an outcome to be associated with some action in the model. Action outcomes associate Boolean values to actions in the test and maintenance process.

EXPRESS specification:

```
*)
  ENTITY ActionOutcome
    SUBTYPE OF(Outcome);
  WHERE
    validOutcomes                    :SELF\Outcome.allowedValue IN
                    [DONE, ABORTED, NOTSTARTED, USERDEFINED, UNKNOWN];
  END_ENTITY;
(*
```

Formal propositions:

validOutcomes	Ensures the actual outcome value corresponds to one of the outcomes in the constrained list.

B.1.40 ClassifiedAttributes

This shall be used by all entities that require security classification identification.

EXPRESS specification:

```
*)
  ENTITY ClassifiedAttributes;
    classified                      :BOOLEAN;
    securityClassification          :OPTIONAL ClassificationType;
  WHERE
    populatedClassification         :NOT(SELF.classified)OR
                        EXISTS(SELF.securityClassification);
  END_ENTITY;
(*
```

Attribute definitions:

classified : An indication that the element is or is not classified. If set to true, then the element is classified.

securityClassification : A user-defined string declaring the highest security classification level of the classified element.

Formal propositions:

populatedClassification Requires that if the element is classified, a security classification has been specified for the element.

B.1.41 Collection

This entity aggregates multiple data values, i.e., unordered sets, ordered vectors, or other collections of named values.

EXPRESS specification:

```
*)
  ENTITY Collection
    SUBTYPE OF(ValueType);
    Members             :BAG [1:?] OF Item;
  END_ENTITY;
(*
```

Attribute definitions:

members : The group of items making up the collection.

B.1.42 CommonValue

Defines an entity from the Common schema that specifies value information. This entity is a supertype of a NamedValue where the distinction is that the CommonValue does not include a named identifier. The

CommonValue will be instantiated either as a single data item (whose type is determined when instantiated), a set of CommonValues, or an indexed array of CommonValues.

EXPRESS specification:

```
*)
  ENTITY CommonValue;
    hasValue              :ValueType;
  END_ENTITY;
(*
```

Attribute definitions:

hasValue : Provides the actual instantiation of a value in the document.

B.1.43 Contact

Entity “Contact” includes contact information for an organization or person.

EXPRESS specification:

```
*)
  ENTITY Contact
    SUBTYPE OF(Person);
    email                 :OPTIONAL SET [1:?] OF EmailAddress;
    phone                 :OPTIONAL SET [1:?] OF TelephoneNumber;
  END_ENTITY;
(*
```

Attribute definitions:

email : Attribute “email” identifies a specific person’s email address.

phone : Attribute “phone” provides a person’s telephone number.

B.1.44 DataValue

This entity shall be used to contain a single data value.

EXPRESS specification:

```
*)
  ENTITY DataValue
    SUBTYPE OF(ValueType);
    singleValue           :Datum;
  END_ENTITY;
(*
```

Attribute definitions:

singleValue : Provides a specific individual value for the datum.

B.1.45 DatumQuality

This entity shall be used to record statistics characterizing descriptive properties of the data element.

EXPRESS specification:

```
*)
  ENTITY DatumQuality;
    resolution          :OPTIONAL NumericType;
    errorLimit          :OPTIONAL Limit;
    range               :OPTIONAL Limit;
    confidence          :OPTIONAL NumericType;
  END_ENTITY;
(*
```

Attribute definitions:

resolution : Contains the number of significant digits of the datum.

errorLimit : Specifies the deviation from some nominal value.

range : Specifies bounds for legal values on the datum.

confidence : Specifies the degree of certainty in the validity of the value of the datum. The actual implementation of confidence is implementation specific.

B.1.46 DescriptionDocumentReference

Identifies an external document.

EXPRESS specification:

```
*)
  ENTITY DescriptionDocumentReference;
    documentID          :Identifier;
    uniqueID            :UUID;
  UNIQUE
    oneID               :uniqueID;
  END_ENTITY;
(*
```

Attribute definitions:

documentID : The unique identifier for the document.

uniqueID : The universally unique identifier for the document.

B.1.47 Document

This entity shall be used to capture identification information for a document or to point directly to an instance of another relevant standard instance of a document (e.g., IEEE Std 1232 or IEEE Std 1636.2).

EXPRESS specification:

```
*)
  ENTITY Document;
    uniqueID                          : UUID;
    documentName                      : NameType;
    documentControlNumber             : OPTIONAL SequenceCode;
    documentVersion                   : OPTIONAL Version;
    actualDocument                    : OPTIONAL SiteOrText;
  UNIQUE
    oneUUID :uniqueID;oneControl      : documentControlNumber;
END_ENTITY;
(*
```

Attribute definitions:

uniqueID : Provides the universally unique identifier for the instance document. If the subtypes are instantiated, the identification mechanism for that subtype is used. For example, for the AI-ESTATE document, the UUID will actually be an OID.

documentName : Provides a text-based name (e.g., title) of a document.

documentControlNumber : A unique organization-specific identifier for the document.

documentVersion : Attribute “documentVersion” defines a string type to provide the version of a particular document.

actualDocument : Provides access to the text of the actual document, either by means of a web site or by embedding the actual text.

B.1.48 Expected

Contains the expected value of the associated limit or mask.

EXPRESS specification:

```
*)
  ENTITY Expected;
    comparator          :EqualityComparatorType;
    expectedValue       :CommonValue;
  END_ENTITY;
(*
```

Attribute definitions:

comparator : The comparison operator (equality or inequality) to be applied when comparing to the limit value.

expectedValue : The instantiation of the specific expected value.

B.1.49 HardwareInstance

An entity containing information to characterize any hardware item.

EXPRESS specification:

```
*)
  ENTITY HardwareInstance
    SUBTYPE OF(SystemInstance);
    calibrationDate     :OPTIONAL DateTime;
    usage               :OPTIONAL PowerOn;
  END_ENTITY;
(*
```

Attribute definitions:

calibrationDate : Attribute “calibrationDate” provides the date and time at which the hardware item was last calibrated.

usage : Attribute “usage” is an optional parameter recorded by a maintainer indicating usage information about the UUT. Here, usage is defined to be elapsed time and number of power on/off cycles.

B.1.50 Identification

Entity “Identification” defines the information needed to identify some item, including a specific identifier and information on the type of identifier.

EXPRESS specification:

```
*)
  ENTITY Identification;
    designator          :OPTIONAL AbbreviationType;
    supplierVersion     :OPTIONAL Version;
    modelName           :NameType;
    idNumbers           :SET[1:?] OF IdentificationNumbers;
    manufacturers       :SET[1:?] OF Organization;
  END_ENTITY;
(*
```

Attribute definitions:

designator : Alphanumeric string that identifies an item in a larger assembly.

supplierVersion : A textual description providing the version of the item.

modelName : Contains the model name of the item.

idNumbers : Attribute “idNumbers” identifies multiple part or model numbers for the described item.

manufacturers : Attribute “manufacturer” identifies a specific manufacturer of an item.

B.1.51 IdentificationNumber

Entity “IdentificationNumber” provides for multiple end user-assigned part or model numbers for the described item.

EXPRESS specification:

```
*)
  ENTITY IdentificationNumber;
    idNumber             :Identifier;
    idType               :IdentifierType;
    qualifier            :DescriptionType;
  END_ENTITY;
(*
```

Attribute definitions:

idNumber : The identification number of the entity.

idType : An indication of whether this is a part number, a model number, or some other type of identifier.

qualifier : Attribute “qualifier” provides additional descriptive data that further specifies or identifies the number attribute.

B.1.52 IdentificationNumbers

Entity “IdentificationNumbers” is a collector for an unbounded set of identification numbers or manufacturer identification numbers. This identifies multiple part or model numbers for the described item (such as a user or manufacturer part or model number).

EXPRESS specification:

```
*)
  ENTITY IdentificationNumbers;
    ids                  :SET [1:?] OF IdChoice;
  END_ENTITY;
(*
```

Attribute definitions:

ids : The set of actual identifiers.

B.1.53 IndexedArray

Entity “IndexedArray” shall contain multi-dimensional arrays of data. This information includes simple name-value pairs as well as more complex matrices.

EXPRESS specification:

```
*)
  ENTITY IndexedArray
    SUBTYPE OF(ValueType);
    members              :LIST [1:?] OF DataValue;
    dimension            :StringType;
  WHERE
    validDim             :ValidDimensions(SELF.dimension) = TRUE;
  END_ENTITY;
(*
```

Attribute definitions:

members : The list of actual data values contained within the array.

dimension : A string designating an n-dimensional array index of the format [a, b, c, ..., n].

Formal propositions:

validDim Tests the specified array dimension, represented as a string, to ensure it is specified correctly.

B.1.54 InstallAction

Entity “InstallAction” is a subtype of MaintenanceAction and corresponds to the act of installing either a repaired or a new component into a unit to complete the repair.

EXPRESS specification:

```
*)
  ENTITY InstallAction
    SUBTYPE OF(MaintenanceAction);
    itemInstalled        :SystemInstance;
  END_ENTITY;
(*
```

Attribute definitions:

itemInstalled : Attribute “itemInstalled” identifies the specific component being installed back into a unit to complete the maintenance action.

B.1.55 Item

Entity Item contains an individual data value or vector.

EXPRESS specification:

```
*)
  ENTITY Item;
    name                :NameType;
    itemChoice          :OPTIONAL ValueType;
  END_ENTITY;
(*
```

Attribute definitions:

name : A descriptive or common name for the individual data value or vector.

itemChoice : An item may be instantiated by a single value, a recursive collection of values, or an indexed array of values.

B.1.56 ItemDescription

Entity “ItemDescription” is a document providing a technical description of the item being maintained. An example of an item description document is based on the UUT Description provided by IEEE Std 1671.3.

EXPRESS specification:

```
*)
  ENTITY ItemDescription;
    procurerVersion     :OPTIONAL Version;
    name                :NameType;
    description         :OPTIONAL DescriptionType;
    itemIdentifier      :Identification;
  END_ENTITY;
(*
```

Attribute definitions:

procurerVersion : Attribute “procurerVersion” defines a string type to provide the version of a particular document.

name : Attribute “name” provides that name of the document.

description : Attribute “description” provides freeform text describing the document.

itemIdentifier : Attribute “itemIdentifier” provides detailed identification information for the class of item/system.

B.1.57 ItemDesignReference

Supports either describing an item by an explicit item description or an external document.

EXPRESS specification:

```
*)
  ENTITY ItemDesignReference;
    definitionOrDocRef  :DesignDescriptor;
  END_ENTITY;
(*
```

Attribute definitions:

definitionOrDocRef : Attribute “definitionOrDocRef” chooses between information on the technical specifications and manufacturer information about the design of a maintenance item.

B.1.58 ItemInstance

Entity “ItemInstance” is a document providing a technical description of the serial numbered item being maintained. An example of an item description document is based on the UUT description provided by IEEE Std 1671.3.

EXPRESS specification:

```
*)
  ENTITY ItemInstance
    SUBTYPE OF(ItemDesignReference);
    serialNumber        :Identifier;
  UNIQUE
    oneSerialNo         :serialNumber;
  END_ENTITY;
(*
```

Attribute definitions:

serialNumber : Attribute “serialNumber” provides the unique identifying serial number for the maintenance item.

B.1.59 ItemInstanceReference

Supports either describing an actual item by either an explicit item description or an external document.

EXPRESS specification:

```
*)
  ENTITY ItemInstanceReference;
    instanceOrDoc       :InstanceDescriptor;
  END_ENTITY;
(*
```

Attribute definitions:

instanceOrDoc : Attribute “instanceOrDoc” chooses between information on the technical specifications and manufacturer information about the specific maintenance item.

B.1.60 Limit

An entity that provides information on the extent or range of some value. “Limit” is defined as a choice between a desired or expected value, a pair of limit values, or a numeric mask value. If more than one “Limit” child element exists, then the logical expression that combines all limits shall be constructed by taking the first limit (without its operator) and then appending subsequent limits in the order in which they appear in the instance document, each prefixed by its operator.

EXPRESS specification:

```
*)
  ENTITY Limit;
    andOrOperator       :OPTIONAL LogicalOperator;
    limitName           :OPTIONAL NameType;
    particularLimit     :LimitType;
    description         :OPTIONAL DescriptionType;
  WHERE
    and_Or              :SELF.andOrOperator <> 'XOR';
  END_ENTITY;
(*
```

Attribute definitions:

andOrOperator : Specifies an operator corresponding either to a logical AND or a logical OR.

limitName : A descriptive or common name for the limit expressed in the element.

particularLimit : Chooses between an expected value, a single limit, a mask, or a limit pair.

description : Textual description of a limit.

Formal propositions:

and_Or Ensures the logical operator is not XOR.

B.1.61 LimitPair

Entity “LimitPair” shall contain the pair of values that bound the limit range.

EXPRESS specification:

```
*)
  ENTITY LimitPair;
    andOrOperator       :OPTIONAL LogicalOperator;
    pairName            :OPTIONAL NameType;
    pair                :LIST [2:2] OF SingleLimit;
    nominal             :OPTIONAL CommonValue;
  WHERE
    and_Or              :SELF.andOrOperator <> 'XOR';
  END_ENTITY;
(*
```

Attribute definitions:

andOrOperator : Specifies an operator corresponding either to a logical AND or a logical OR.

pairName : A description or common name for the limit pair expressed in the element.

pair : Contains two and only two single limits.

nominal : Contains the expected or preferred value to be captured.

Formal propositions:

and_Or Ensures the logical operator is not XOR.

B.1.62 MailingAddress

Entity “MailingAddress” specifies an address for sending information through a government or private postal service.

EXPRESS specification:

```
*)
  ENTITY MailingAddress;
    address1             :AddressComponent;
    address2             :OPTIONAL AddressComponent;
    city                 :AddressComponent;
    state                :AddressComponent;
    country              :AddressComponent;
    postalCode           :AddressComponent;
  END_ENTITY;
(*
```

Attribute definitions:

address1 : Attribute “address1” provides the first line of a mailing address.

address2 : Attribute “address2” provides the second line of a mailing address.

city : Attribute “city” provides the city of a mailing address.

state : Attribute “state” provides the state or province of a mailing address.

country : Attribute “country” provides the country of a mailing address.

postalCode : Attribute “postalCode” provides the zip code or postal code of a mailing address.

B.1.63 MaintenanceAction

Entity “MaintenanceAction” defines a specific action taken in the maintenance of some item or entity.

EXPRESS specification:

```
*)
  ENTITY MaintenanceAction
    ABSTRACT    SUPERTYPE    OF    (ONEOF(RemoveAction,    RepairAction,
            InstallAction, NoAction));
    actionID             :Identifier;
    name                 :NameType;
    startTime            :DateTime;
    endTime              :OPTIONAL DateTime;
    totalTime            :Duration;
    preventative         :BOOLEAN;
    actionCode           :OPTIONAL Code;
    description          :DescriptionType;
  UNIQUE
    oneID                :actionID;
  END_ENTITY;
(*
```

Attribute definitions:

actionID : Attribute “actionID” provides a unique identifier for the maintenance action.

name : Attribute “name” provides a user-friendly textual name for the maintenance action.

startTime : Attribute “startTime” provides the date and time for the initiation of a maintenance event.

endTime : Attribute “endTime” provides the date and time for the termination of a maintenance event.

totalTime : Attribute “totalTime” provides the aggregate personnel time for the actual work duration of a maintenance event.

preventative : Attribute “preventative” identifies whether or not the MaintenanceAction was performed as part of preventative maintenance (true) or not (false).

actionCode : Attribute “actionCode” provides the code that best describes the maintenance action performed. Maintenance organizations typically have a list of organization-unique codes that best describe the reasons for maintenance action performed. This attribute provides a place to record that code. The meaning and interpretation of the code is implementation-specific.

description : Attribute “description” shall be used to provide a detailed textual description of the maintenance action performed.

B.1.64 MaintenanceLevel

Entity “MaintenanceLevel” is used to identify and describe a level of maintenance in a maintenance architecture. Common examples include organizational, intermediate, depot, and original equipment manufacturer.

EXPRESS specification:

```
*)
  ENTITY MaintenanceLevel;
    levelAbbreviation   :AbbreviationType;
    levelName           :OPTIONAL NameType;
  END_ENTITY;
(*
```

Attribute definitions:

levelAbbreviation : Attribute "levelAbbreviation" specifies the abbreviation (e.g., O, I, D, or OEM) of the level of maintenance.

levelName : Attribute "levelName" specifies a descriptive or common name for the level of maintenance. Examples: Intermediate and Depot.

B.1.65 ManufacturerIdNumber

Entity "ManufacturerIdNumber" is a subtype of IdentificationNumber and provides identification information specific to manufacturers of some item.

EXPRESS specification:

```
*)
  ENTITY ManufacturerIdNumber;
    idNumber            :Identifier;
    idType              :IdentifierType;
    manufacturerName    :DescriptionType;
  END_ENTITY;
(*
```

Attribute definitions:

idNumber : A number used to identify the entity as specified by the manufacturer of that entity.

idType : An indication of whether this is a part number, a model number, or some other type of identifier.

manufacturerName : A descriptive or common name for the manufacturer.

B.1.66 Mask

Entity "Mask" shall contain a numeric mask value, including the value type, pattern, and masking operation on that value.

EXPRESS specification:

```
*)
  ENTITY Mask;
    expectedValue        :Expected;
    maskedValue          :LIST [1:?] OF CommonValue;
    name                 :OPTIONAL NameType;
    maskOperator         :LogicalOperator;
  END_ENTITY;
(*
```

Attribute definitions:

expectedValue : Contains the expected pattern.

maskedValue : Contains the mask pattern.

name : A descriptive or common name for the limit.

maskOperator : Specifies an operator corresponding either to a logical AND, a logical OR, or an exclusive OR.

B.1.67 NamedValue

Entity “NamedValue” contains a data value with which a textual name must be associated.

EXPRESS specification:

```
*)
  ENTITY NamedValue
    SUBTYPE OF(CommonValue);
    name                 : NameType;
  END_ENTITY;
(*
```

Attribute definitions:

name : A descriptive or common name for the value.

B.1.68 NoAction

Entity “NoAction” is a subtype of MaintenanceAction and corresponds to information about why no action was taken to repair a unit.

EXPRESS specification:

```
*)
  ENTITY NoAction
    SUBTYPE OF(MaintenanceAction);
    SELF\MaintenanceAction.description          :DescriptionType;
  END_ENTITY;
(*
```

Attribute definitions:

description : Attribute “description” provides the reason for not being able to perform maintenance on the unit.

B.1.69 Organization

Entity “Organization” defines key identifying and descriptive information about a organization used to support, repair, or otherwise maintain systems.

EXPRESS specification:

```
*)
  ENTITY Organization;
    name                :NameType;
    cageCode            :OPTIONAL Identifier;
    address             :OPTIONAL MailingAddress;
    contacts            :OPTIONAL SET [1:?] OF Contact;
    faxNumber           :OPTIONAL TelephoneNumber;
    webAddress          :OPTIONAL URL;
    workCenter          :OPTIONAL NameType;
  UNIQUE
    oneID               :cageCode;
  END_ENTITY;
(*
```

Attribute definitions:

name : Attribute ‘name’ provides the name of the manufacturing or maintenance facility.

cageCode : Attribute “cageCode” provides a unique identifier for the maintenance organization. A common identifier used in the USA and in NATO countries is the commercial and government entity (CAGE) code.

address : Attribute “address” provides the mailing address of the organization.

contacts : Attribute “contacts” provides detailed identification and contact information about one or more points of contact for the organization.

faxNumber : Attribute “faxNumber” provides a facsimile number for the organization.

webAddress : Attribute “webAddress” provides the URL for the facility web site.

workCenter : The manufacturing or maintenance facility in which information was collected.

B.1.70 Outcome

This simple type enumerates the text values defined in this standard to represent permitted values for an outcome. It is a supertype of testOutcome and actionOutcome.

EXPRESS specification:

```
*)
  ENTITY Outcome
    ABSTRACT SUPERTYPE OF (ONEOF(ActionOutcome, TestOutcome));
    allowedValue        :OutcomeValues;
    qualifier           :OPTIONAL QualifierType;
    forced              :OPTIONAL BOOLEAN;
    referenceID         :OPTIONAL Identifier;
  END_ENTITY;
(*
```

Attribute definitions:

allowedValue : Attribute ‘allowedValue’ defines a value that can belong to the specific outcome based on the subtype instantiating the Outcome entity.

qualifier : An adjective providing additional descriptive data for the ‘allowedValue’ attribute.

forced : Attribute “forced” indicates whether the recorded outcome is a user “override” of the observed outcome for the test.

referenceID : A reference to the test program that generated the outcome.

B.1.71 Person

Entity “Person” specifies identifying information about a person involved in the maintenance of a system.

EXPRESS specification:

```
*)
  ENTITY Person;
    ID                  :Identifier;
    Name                :NameType;
    otherData           :OPTIONAL NamedValue;
    affiliation         :OPTIONAL Organization;
    address             :OPTIONAL MailingAddress;
  UNIQUE
    oneID               :ID;
  END_ENTITY;
(*
```

Attribute definitions:

ID : Provides a unique identifier for a specific person.

name : Attribute “name” identifies the name of the person performing the maintenance action.

otherData : Contains information associated with the subject person other than that provided elsewhere with this entity.

affiliation : Attribute “affiliation” identifies the organizational affiliation of a particular person.

address : Attribute “address” provides the mailing addresses for a particular person.

B.1.72 PowerOn

Entity “PowerOn” defines information related to when power is applied to a piece of hardware, including the number of power on/off cycles.

EXPRESS specification:

```
*)
  ENTITY PowerOn;
    cycles              :NUMBER;
    cumulativeTime      :Duration;
  END_ENTITY;
(*
```

Attribute definitions:

cycles : Attribute “cycles” indicates the number of power on/off cycles experienced by the hardware item at the time of record creation. This information may be useful in applications where diagnostics are performed.

cumulativeTime : Attribute “cumulativeTime” is the total cumulative time that power has been applied to the described hardware item. This information may be useful in applications where diagnostics are performed.

B.1.73 ReferenceDesignator

Entity “ReferenceDesignator” is an unambiguous identifier within the context of a system to include the kind of part (e.g., resistor, pump, etc.).

EXPRESS specification:

```
*)
  ENTITY ReferenceDesignator;
    abbreviation        :AbbreviationType;
    indictedPartType    :TypeDescription;
    description         :OPTIONAL DescriptionType;
    failureModes        :OPTIONAL SET [1:?] OF FaultID;
  END_ENTITY;
(*
```

Attribute definitions:

abbreviation : Attribute “abbreviation” provides a short-hand label for the reference designator.

indictedPartType : Attribute “indictedPartType” provides the type of part being indicted within the UUT, e.g., WRA, SRA, resistor, actuator, etc.

description : The description that corresponds to the reference designator.

failureModes : Attribute “failureModes” identifies the specific failure modes associated with the component indicated by the current reference designator.

B.1.74 RemoveAction

Entity “RemoveAction” is a subtype of a MaintenanceAction and corresponds to removal of a component to effect a maintenance action.

EXPRESS specification:

```
*)
  ENTITY RemoveAction
    SUBTYPE OF(MaintenanceAction);
    itemRemoved          : SystemInstance;
  END_ENTITY;
(*
```

Attribute definitions:

itemRemoved :Attribute “itemRemoved” identifies the maintenance item for subsequent repair.

B.1.75 RepairAction

Entity “RepairAction” is a subtype of MaintenanceAction and provides information on the work performed to restore a part to an operational state.

EXPRESS specification:

```
*)
  ENTITY RepairAction
    SUBTYPE OF(MaintenanceAction);
    itemRepaired         :SystemInstance;
    repairType           :RepairCode;
  END_ENTITY;
(*
```

Attribute definitions:

itemRepaired : Attribute “itemRepaired” identifies the specific unit undergoing repair action.

repairType : A code corresponding to the type of repair completed.

B.1.76 SingleLimit

Entity SingleLimit contains the value being used as a threshold for purposes of single value comparison.

EXPRESS specification:

```
*)
  ENTITY SingleLimit;
    comparator           :InequalityComparatorType;
    limtValue            :CommonValue;
  END_ENTITY;
(*
```

Attribute definitions:

comparator : The comparison operator (less than, greater than, etc.) to be applied when comparing to the limit value.

limtValue : Contains the value of the threshold to be compared.

B.1.77 SoftwareInstance

Entity “SoftwareInstance” supports the case where the object of maintenance is a software component rather than a hardware component.

EXPRESS specification:

```
*)
  ENTITY SoftwareInstance
    SUBTYPE OF(SystemInstance);
  END_ENTITY;
(*
```

B.1.78 SystemInstance

Entity “SystemInstance” defines a specific target of maintenance against which the maintenance action has been performed.

EXPRESS specification:

```
*)
  ENTITY SystemInstance
    ABSTRACT SUPERTYPE OF (ONEOF(HardwareInstance, SoftwareInstance))
    SUBTYPE OF(ItemInstance);
    issueDate           :OPTIONAL DateTime;
    itemWarranty        :OPTIONAL SET [1:?] OF Warranty;
  UNIQUE
    oneSerialNo         :serialNumber;
  END_ENTITY;
(*
```

Attribute definitions:

issueDate : Attribute “issueDate” provides the date the maintenance instance was issued as available for use. This date is applied to both hardware (as a manufacture date) or software (as a software release date).

itemWarranty : Attribute “itemWarranty” identifies the terms of a set of warranties on the maintenance item. More than one warranty is permitted to support multiple warranty such as those provided by a manufacturer, a maintainer, a supplement, or an extension.

B.1.79 TestOutcome

This simple type enumerates the text values defined in this standard to represent permitted values for a test outcome. The enumerations defined are based on information gathered specifically for preparation of this

standard. In use, “Passed” shall indicate the results of a test were within specified limits; “Failed” shall indicate the opposite. “Aborted” shall be used for any test that did not complete normally.

EXPRESS specification:

```
*)
  ENTITY TestOutcome
    SUBTYPE OF(Outcome);
  WHERE
    validOutcomes       : SELF\Outcome.allowedValue IN
            [PASSED,    FAILED,    ABORTED,    NOTSTARTED,    USERDEFINED,
            NOTKNOWN];
END_ENTITY;
(*
```

B.1.80 ValueType

Entity ValueType contains typed values. Different elements shall be used to represent a single data value, a collection of data values, or an array of data values.

EXPRESS specification:

```
*)
  ENTITY ValueType
    ABSTRACT SUPERTYPE OF (ONEOF(DataValue, Collection, IndexedArray));
    defaultStandardUnit             :OPTIONAL StandardUnit;
    defaultNonStandardUnit          :OPTIONAL NonStandardUnit;
    defaultUnitQualifier            :OPTIONAL StringType;
    quality                         :DatumQuality;
  END_ENTITY;
(*
```

Attribute definitions:

defaultStandardUnit : When used, this attribute shall contain only a unit of measure defined in IEEE Std 260.1.

defaultNonStandardUnit : When used, this attribute shall contain any unit not defined by IEEE Std 260.1.

defaultUnitQualifier : When used, this provides a textual qualifier that modifies the semantics of a standard or non-standard unit (e.g., RMS or Peak-to-Peak).

quality : The quality characteristics of the data item.

B.1.81 Warranty

Entity “Warranty” defines the manufacturer’s legal responsibility for the support of a given system or item.

EXPRESS specification:

```
*)
  ENTITY Warranty;
    warrantedBy                    :Organization;
    warrantyDuration               :Duration;
    warrantyEffectiveDate          :Date;
    warrantyExclusion              :SET [1:?] OF Exclusion;
  END_ENTITY;
(*
```

Attribute definitions:

warrantedBy : Name of the manufacturer which warrants the unit.

warrantyDuration : Attribute “warrantyDuration” provides the total time the unit is covered by the warranty.

warrantyEffectiveDate : Attribute “warrantyEffectiveDate” provides the date at which a manufacturer’s warranty begins to take effect.

warrantyExclusion : Attribute “warrantyExclusion” identifies discrepancies in a unit that are not covered by the warranty.

B.1.82 WorkOrder

Entity WorkOrder contains information characterizing the authorization of the testing or maintenance of a system instance.

EXPRESS specification:

```
*)
  ENTITY WorkOrder;
    documentControlNumber          :Identifier;
    maintenanceControlNumber       :OPTIONAL Identifier;
    level                          :OPTIONAL MaintenanceLevel;
    description                    :OPTIONAL DescriptionType;
  END_ENTITY;
(*
```

Attribute definitions:

documentControlNumber : Attribute “workOrderNumber” identifies the instance document or collection of information for the work order, across multiple levels of maintenance.

maintenanceControlNumber : Attribute “workItemNumber” identifies the instance document or collection of information for a work order at a particular level of maintenance.

level : Attribute “level” identifies the maintenance level for this work order.

description : Attribute “description” provides narrative for the work order.

B.1.83 ValidDimensions

A function to ensure that the dimensions on an indexed array are specified properly.

EXPRESS specification:

```
*)
  FUNCTION ValidDimensions
      (dim:StringType):BOOLEAN;
       LOCAL
          i : INTEGER;
       END_LOCAL;
       if (length(dim) <= 2) then
          return(FALSE);
       end_if;
       if (dim[1] <> '[') then
          return(FALSE);
       end_if;
       if (dim[length(dim)] <> ']') then
          return(FALSE);
       end_if;
       if (dim[length(dim)-1] = ',') then
          return(FALSE);
       end_if;
       repeat i := 2 to length(dim) - 1;
          if(NOT(dim[i]IN(['0','1','2','3','4','5','6','7','8','9'])))
    and
            (dim[i] <> ',') then
            return(FALSE);
          end_if;
          if (dim[i] = ',') and (dim[i+1] = ',') then
             return(FALSE);
          end_if;
       end_repeat;
       return(TRUE);
  END_FUNCTION;
END_SCHEMA;
(*
```

B.2 SIMICACommon model EXPRESS-G diagrams

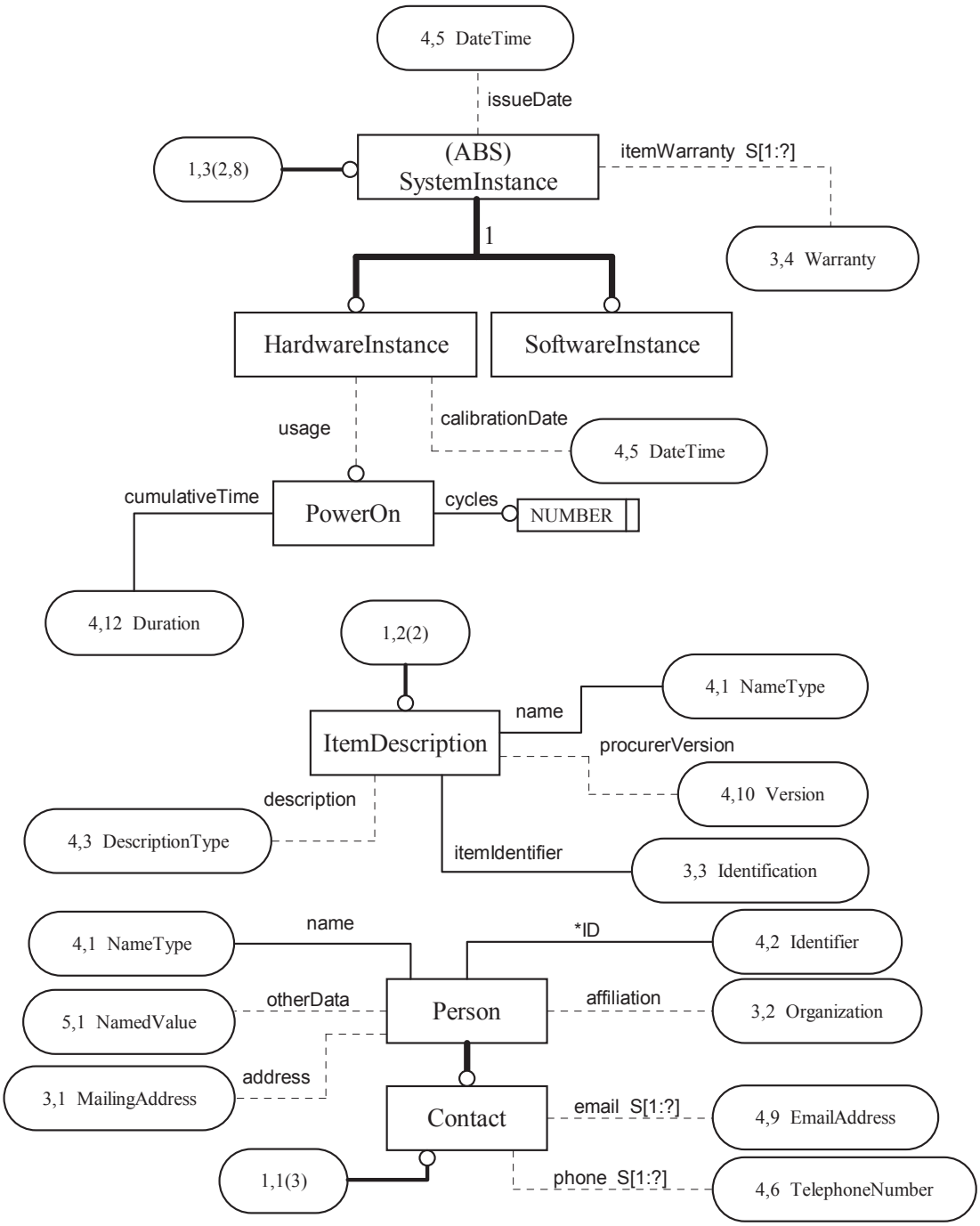

Figure B.1—SIMICA_COMMON_MODEL_DOT_99 EXPRESS-G, diagram 1 of 8

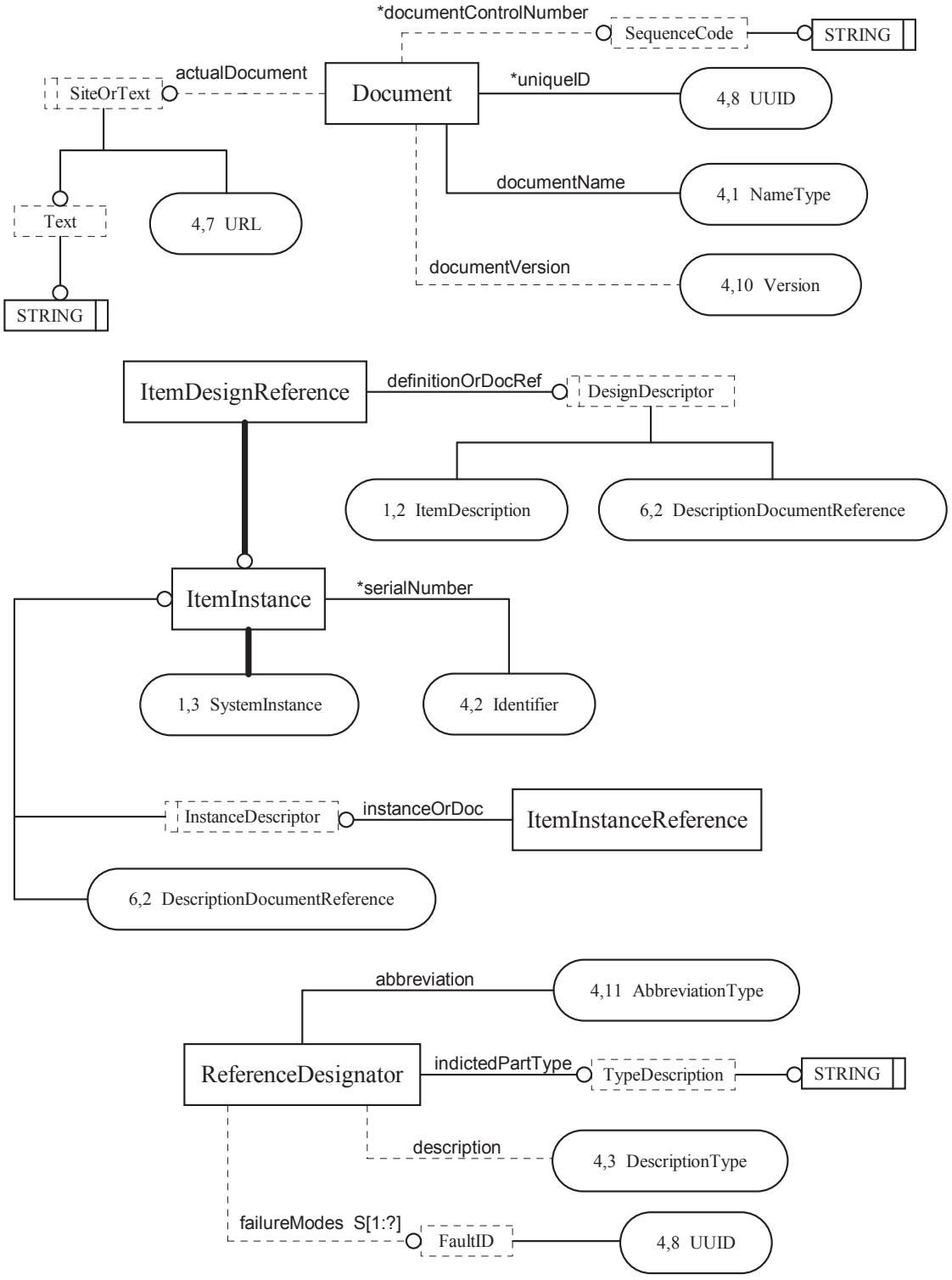

Figure B.2—SIMICA_COMMON_MODEL_DOT_99 EXPRESS-G, diagram 2 of 8

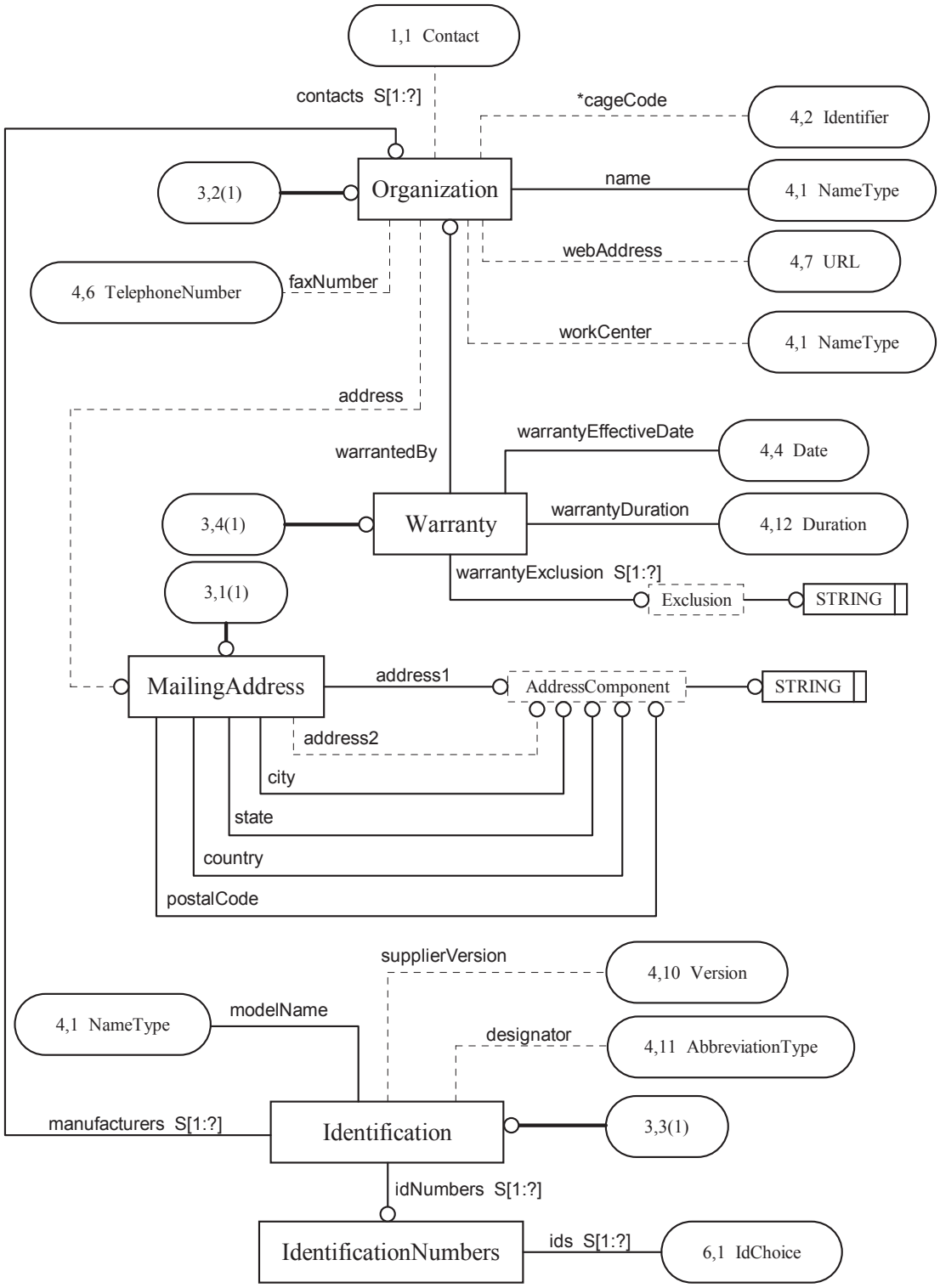

Figure B.3—SIMICA_COMMON_MODEL_DOT_99 EXPRESS-G, diagram 3 of 8

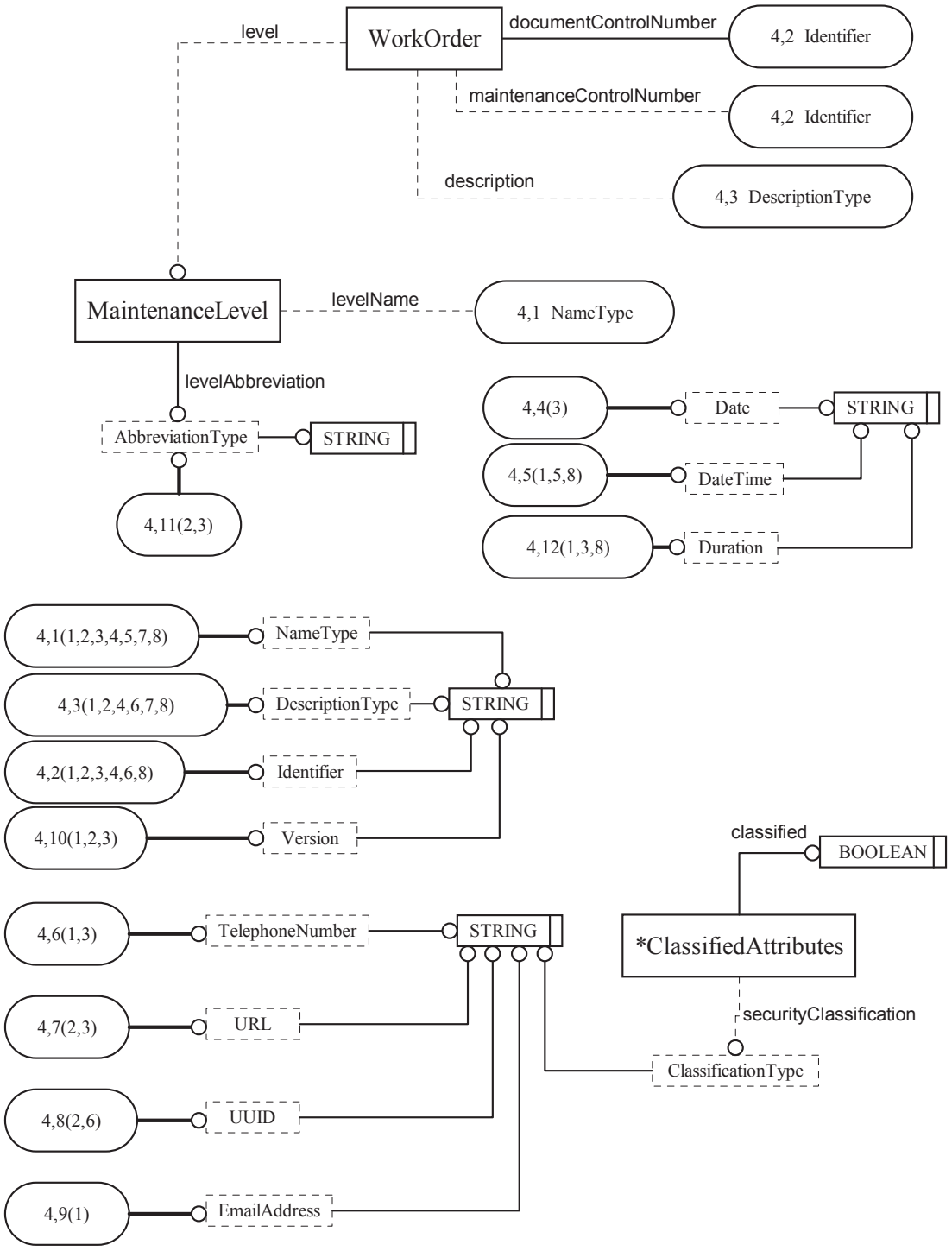

Figure B.4—SIMICA_COMMON_MODEL_DOT_99 EXPRESS-G, diagram 4 of 8

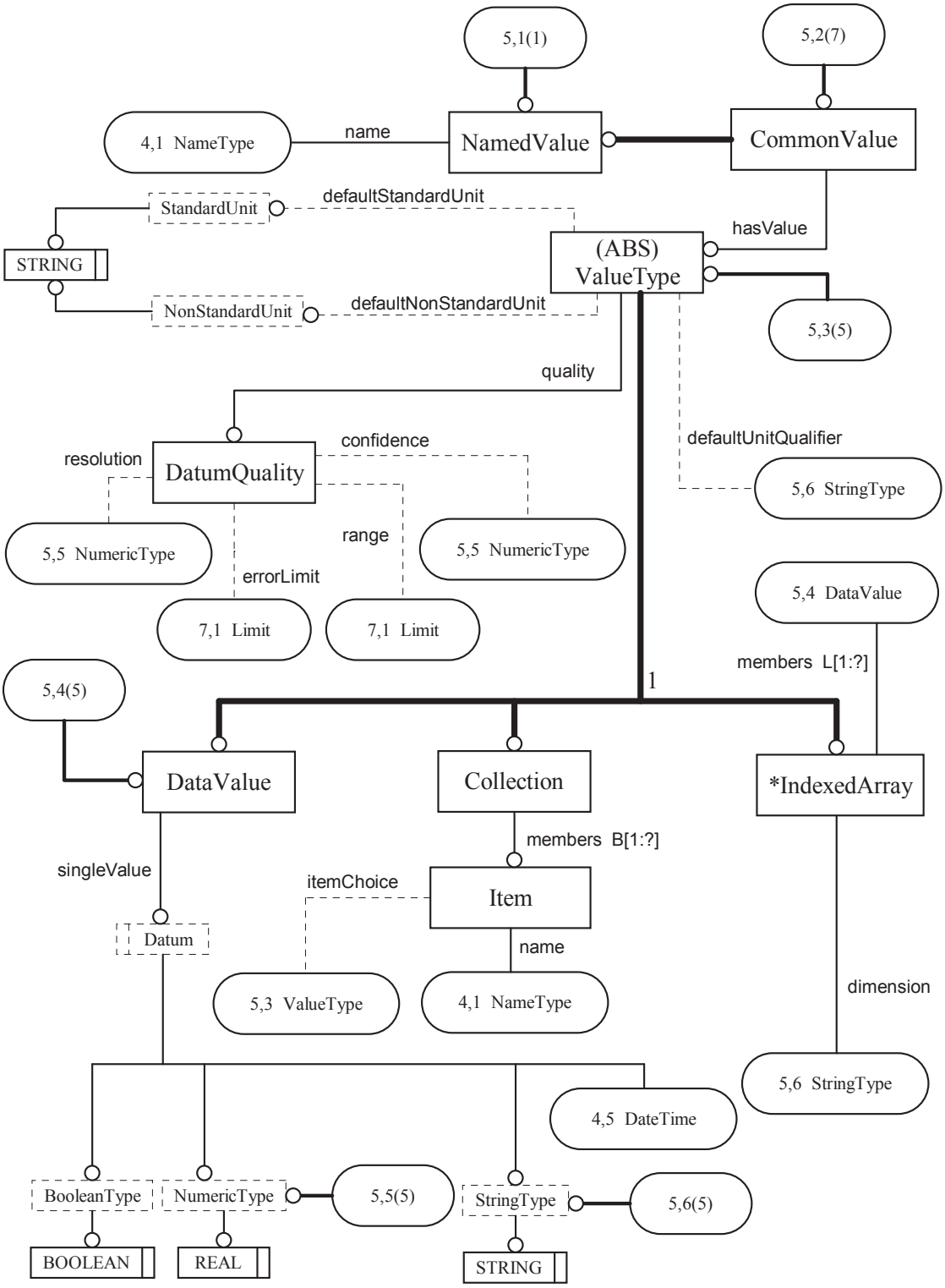

Figure B.5—SIMICA_COMMON_MODEL_DOT_99 EXPRESS-G, diagram 5 of 8

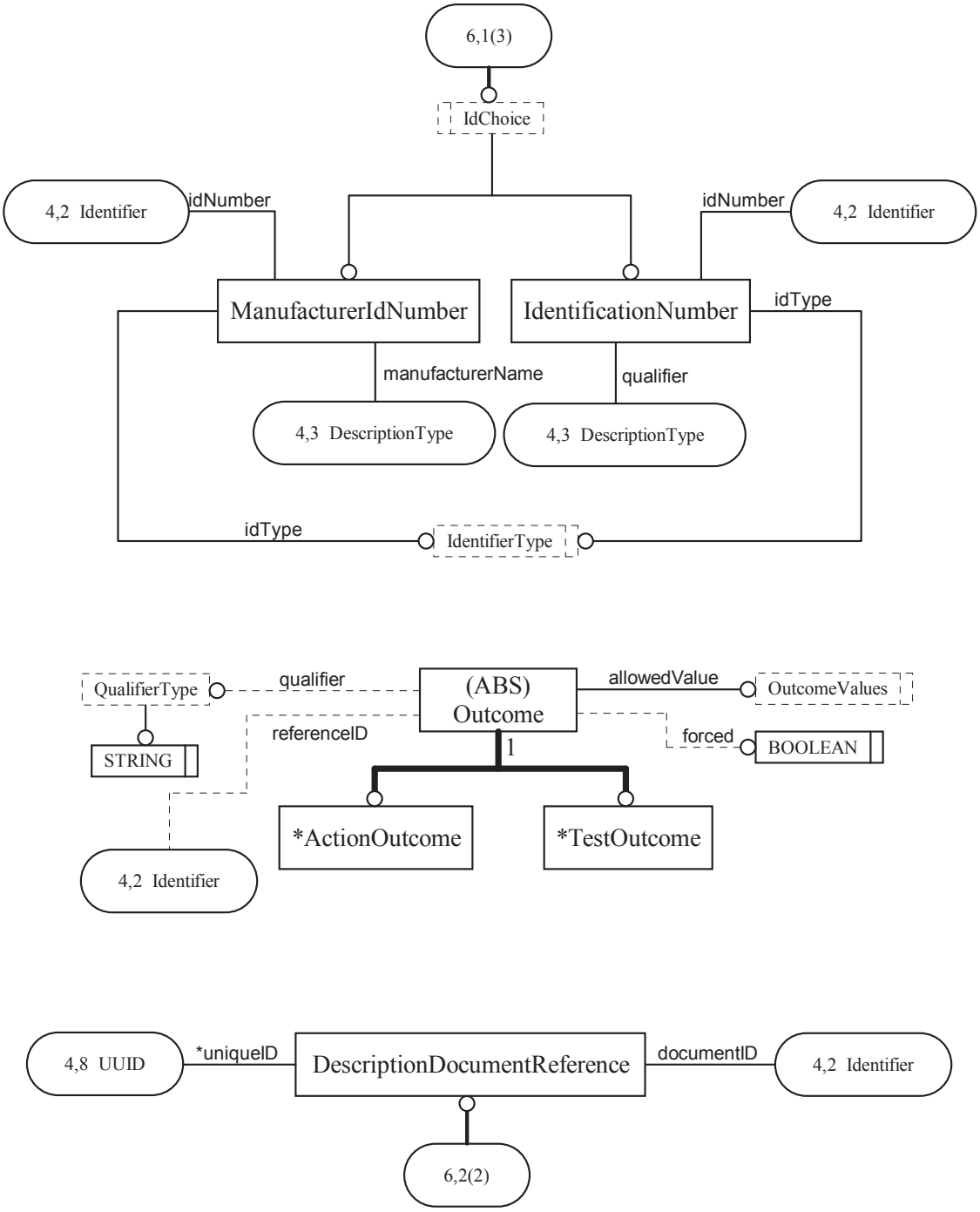

Figure B.6—SIMICA_COMMON_MODEL_DOT_99 EXPRESS-G, diagram 6 of 8

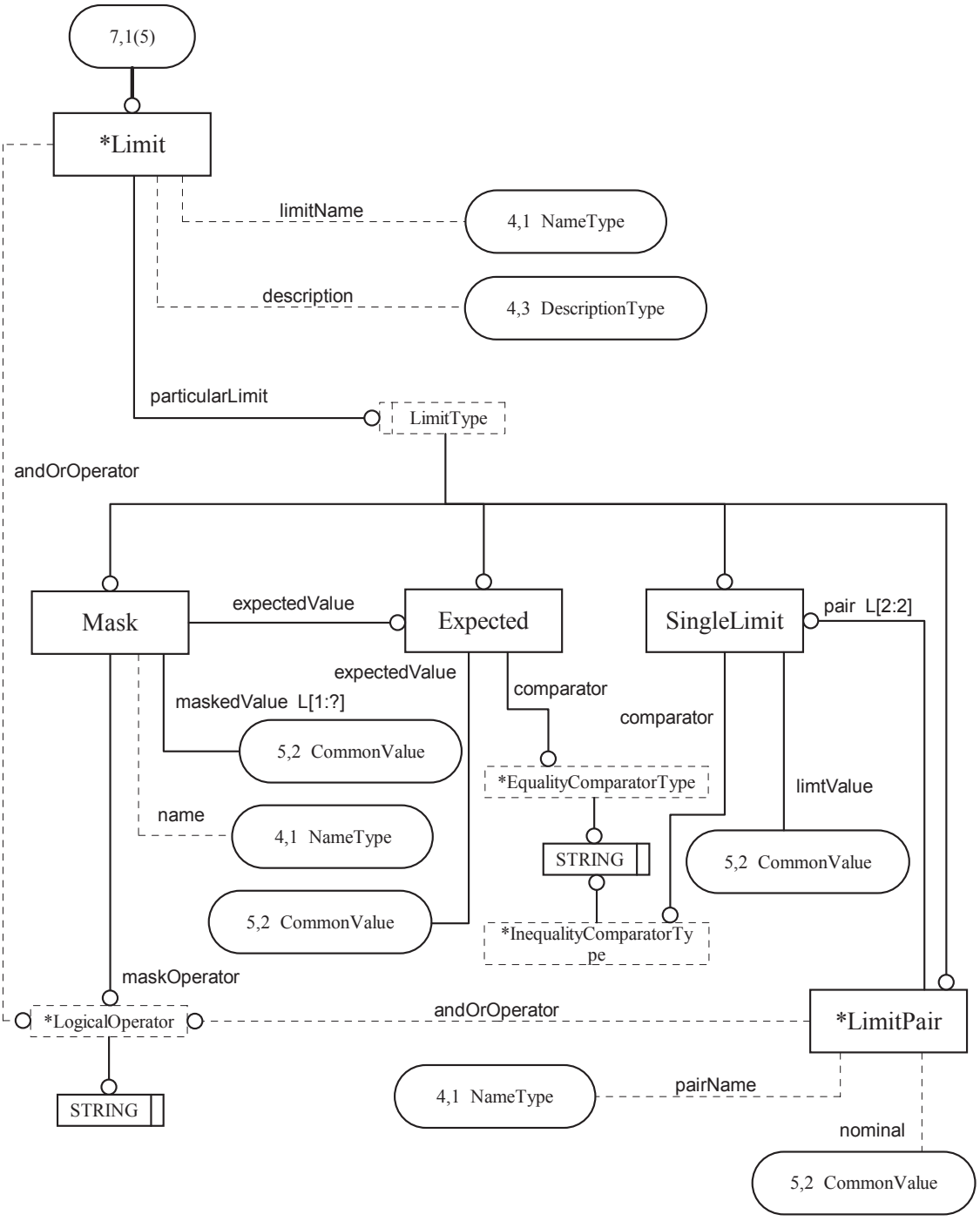

Figure B.7—SIMICA_COMMON_MODEL_DOT_99 EXPRESS-G, diagram 7 of 8

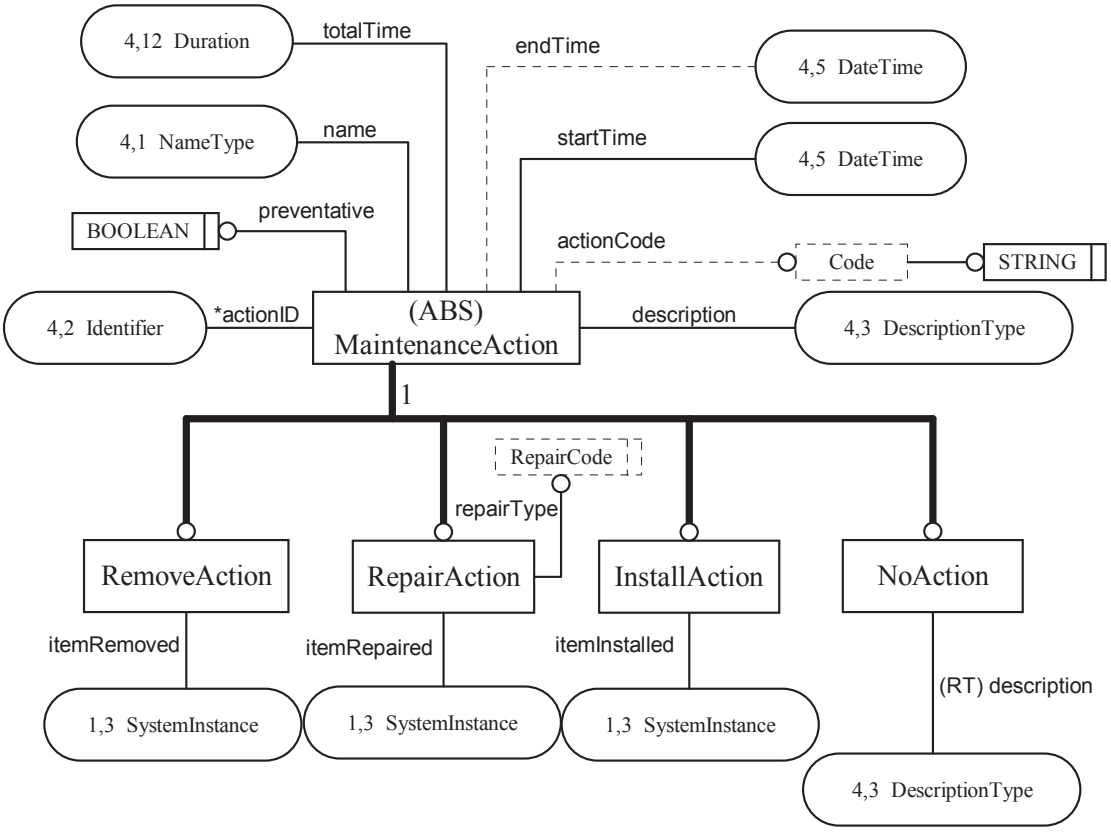

Figure B.8—SIMICA_COMMON_MODEL_DOT_99 EXPRESS-G, diagram 8 of 8

Annex C

(informative)

Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] Extensible Markup Language (XML) 1.0, 5th ed. W3C Proposed Edited Recommendation 05 February 2008.[7]

[B2] *IEEE Standards Dictionary Online*.[8]

[B3] *IEEE Standards Style Manual*.[9]

[B4] IEEE Std 260.1™, IEEE Standard Letter Symbols for Units of Measurement (SI Units, Customary Inch-Pound Units, and Certain Other Units).

[B5] IEEE Std 1232™-2010, IEEE Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE).

[B6] IEEE Std 1636.1™-2007, IEEE Trial-Use Standard for Software Interface for Maintenance Information Collection and Analysis (SIMICA): Exchanging Test Results and Session Information via the Extensible Markup Language (XML).

[B7] IEEE Std 1636.2™-2009, IEEE Trial-Use Standard for Software Interface for Maintenance Information Collection and Analysis (SIMICA): Exchanging Maintenance Action Information via the Extensible Markup Language (XML).

[B8] IEEE Std 1671.1™-2009, IEEE Trial-Use Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML: Exchanging Test Descriptions.

[B9] ISO 10303-11:1994, *Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 11: Description Methods: The EXPRESS Language Reference Manual.*[10]

[B10] Namespaces in XML 1.0, 3rd ed. World Wide Web Consortium Recommendation 8 December 2009. Available: http://www.w3.org/TR/REC-xml-names/.

[B11] Schenk, D. A., and P. R. Wilson, *Information Modeling: The EXPRESS Way*. New York: Oxford University Press, 1994.

[B12] U.S. Navy, *Definitions of Terms for Test, Measurement and Diagnostic Equipment*, MIL-STD-1309D. Washington, DC: Naval Electronics Systems Command (ELEX-8111), 12 February 1992.

[B13] XML Schema Part 0: Primer. Available: http://www.w3.org/TR/xmlschema-0/.

[B14] XML Schema Tutorial. Available: http://www.xfront.com.

[7] Available from World Wide Web: http://www.w3.org/TR/2008/PER-xml-20080205.
[8] *IEEE Standards Dictionary Online* subscription is available at: http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html.
[9] IEEE publications are available from The Institute of Electrical and Electronics Engineers (http://standards.ieee.org/).
[10] ISO publications are available from the ISO Central Secretariat, Case Postalė 56, 1 rue de Varembė, CH-1211, Genevė 20, Switzerland/.Suissė (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/).

[B15] XML Schema Tutorial, Part 1. Available: www.liquid-technologies.com/Tutorials/XmlSchemas/XsdTutorial_01.aspx.

Annex D
(informative)
IEEE list of participants

At the time this IEEE standard was completed, the Diagnostic and Maintenance Control Subcommittee SIMICA CommonWorking Group had the following membership:

Mike Seavey, *Chair*

Chris Gorringe
Teresa Lopes
Ion Neag
John Sheppard
Timothy Wilmering

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

Michael Bodkin
Bill Brown
Malcom Brown
Keith Chow
David Droste
Chris Gorringe
Randall Groves
Werner Hoelzl
Noriyuki Ikeuchi
Anand Jain
Teresa Lopes
Greg Luri
William Maciejewski
Mukund Modi
Charles Ngethe
Leslie Orlidge
Peter Richardson
Robert Robinson
Bartien Sayogo
Mike Seavey
Krishna Seeburn
John Sheppard
Gil Shultz
Joseph Stanco
Walter Struppler
Ronald Taylor
Benton Vandiver
John Vergis
Timothy Wilmering
Oren Yuen
Daidi Zhong

When the IEEE-SA Standards Board approved this standard on 23 August 2013, it had the following membership:

John Kulick, *Chair*
David J. Law, *Vice Chair*
Richard H. Hulett, *Past Chair*
Konstantinos Karachalios, *Secretary*

Masayuki Ariyoshi
Peter Balma
Farooq Bari
Ted Burse
Wael William Diab
Stephen Dukes
Jean-Philippe Faure
Alexander Gelman
Mark Halpin
Gary Hoffman
Paul Houzé
Jim Hughes
Michael Janezic
Joseph L. Koepfinger*
Oleg Logvinov
Ron Petersen
Gary Robinson
Jon Walter Rosdahl
Adrian Stephens
Peter Sutherland
Yatin Trivedi
Phil Winston
Yu Yuan

*Member Emeritus

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

Richard DeBlasio, *DOE Representative*
Michael Janezic, *NIST Representative*

Don Messina
IEEE Standards Program Manager, Document Development

Kathryn Bennett
IEEE Standards Program Manager, Technical Program Development

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch