AEROSPACE STANDARD

AS6522™	REV. B

Issued 2016-12
Revised 2020-08
Stabilized 2024-02

Superseding AS6522A

Unmanned Systems (UxS) Control Segment (UCS) Architecture: Architecture Technical Governance

RATIONALE

This document is no longer part of UCS Architecture as of Rev A.

STABILIZED NOTICE

This document has been declared “STABILIZED” by SAE AS-4UCS Unmanned Systems Control Segment Architecture Committee and will no longer be subjected to periodic reviews for currency. Users are responsible for verifying references and continued suitability of technical requirements. Newer technology may exist.

SAE Executive Standards Committee Rules provide that: “This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user.”

SAE reviews each technical report at least every five years at which time it may be revised, reaffirmed, stabilized, or cancelled. SAE invites your written comments and suggestions.

Copyright © 2024 SAE International

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of SAE.

TO PLACE A DOCUMENT ORDER: **Tel: 877-606-7323 (inside USA and Canada)**
Tel: +1 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org
SAE WEB ADDRESS: http://www.sae.org

For more information on this standard, visit
https://www.sae.org/standards/content/AS6522B/

Documents
Document Sharing Hub

Full document details: SAE AS6522B

INTRODUCTION

This Architecture Technical Governance for the SAE UCS Architecture Model (AS6518) starting at Revision A describes how the Model is structured and how it is to be interpreted, manipulated and extended by users. The UCS Architecture Model is expressed by the Unified Modeling Language® (UML®) and various UML Profiles. Technical Governance describes the purpose of these Profiles and how they are to be applied to the Packages and Diagrams in the Model.

In general terms, the UCS Architecture Model and its user extensions are required to conform to this Technical Governance. UCS Products are in turn required to conform to the UCS Architecture Model and its user extensions. A conformant extension of the UCS Architecture Model can be manipulated by UCS tools. UCS Product conformance is defined in the UCS Architecture Conformance Specification (AS6513).

Technical Governance is expressed through a collection of policies, where each policy describes its motivation, its required usage, constraints and conformance, the policy itself, and application examples of the policy.

APPLICABILITY

This document applies to the UCS Architecture Model (AS6518) and to user extensions of AS6518 for development of conformant UCS Products. This Revision A applies to Revision A (and possibly later revisions) of AS6518.

Documents
Document Sharing Hub
Full document details: SAE AS6522B

TABLE OF CONTENTS

1. SCOPE 6

2. TERMINOLOGY 6

3. REFERENCES 6
3.1 Applicable Documents 6
3.1.1 SAE Publications 6
3.1.2 Object Management Group (OMG) Publications 6
3.1.3 ISO Publications 7
3.1.4 ECMA International Publications 7
3.1.5 U.S. Government Publications 7
3.2 Definitions 7
3.3 Acronyms 7

4. GENERAL POLICIES 8
4.1 Policy on Namespace 8
4.1.1 Policy Usage and Conformance 8
4.1.2 Policy Description and Motivation 8
4.1.3 Policy Constraints 8
4.1.4 Policy 8
4.1.5 Policy Example(s) 8
4.1.6 Policy history 8
4.2 Policy on Model Languages 8
4.2.1 Policy Usage and Conformance 8
4.2.2 Policy Description and Motivation 8
4.2.3 Policy Constraints 8
4.2.4 Policy 9
4.2.5 Policy Examples 9
4.2.6 Policy History 9
4.3 Policy on UCS Architecture Model Package Structure 9
4.3.1 Policy Usage and Conformance 9
4.3.2 Policy Description and Motivation 9
4.3.3 Policy Constraints 9
4.3.4 Policy 9
4.3.5 Policy Example(s) 11
4.3.6 Policy History 11
4.4 General Policy on Naming Conventions 11
4.4.1 Policy Usage and Conformance 11
4.4.2 Policy Description and Motivation 11
4.4.3 Policy Constraints 11
4.4.4 Policy 11
4.4.5 Policy Example(s) 12
4.4.6 Policy History 12

5. INFORMATION MODEL POLICIES 13
5.1 Policy on Conformance to the Data Model Framework 13
5.1.1 Policy Usage and Conformance 13
5.1.2 Policy Description and Motivation 13
5.1.3 Constraints 14
5.1.4 Policy 14
5.1.5 Policy Example(s) 14
5.1.6 Policy History 15
5.2 Policy on Levels of Architecture Abstraction 15
5.2.1 Policy Usage and Conformance 15
5.2.2 Policy Description and Motivation 15
5.2.3 Constraints 15
5.2.4 Policy 15

Documents
Document Sharing Hub
Full document details: SAE AS6522B

5.2.5 Policy Example(s) 15
5.2.6 Policy History 17
5.3 Policy on Conceptual Model Package Structure 17
5.3.1 Policy Usage and Conformance 17
5.3.2 Policy Description and Motivation 17
5.3.3 Policy Constraints 17
5.3.4 Policy 18
5.3.5 Policy Example(s) 18
5.3.6 Policy History 20
5.4 Policy on Conceptual Data Model (CDM) 20
5.4.1 Policy Usage and Conformance 20
5.4.2 Policy Description and Motivation 20
5.4.3 Policy Constraints 20
5.4.4 Policy 20
5.4.5 Policy Example(s) 25
5.4.6 Policy History 26
5.5 Policy on Conceptual Message Model 26
5.5.1 Policy Usage and Conformance 26
5.5.2 Policy Description and Motivation 27
5.5.3 Policy Constraints 27
5.5.4 Policy 27
5.5.5 Policy Example(s) 28
5.5.6 Policy History 31
5.6 Policy on Conceptual Service Model 31
5.6.1 Policy Usage and Conformance 31
5.6.2 Policy Description and Motivation 31
5.6.3 Policy Constraints 31
5.6.4 Policy 32
5.6.5 Policy Example(s) 33
5.7 Policy on Logical Model Package Structure 35
5.7.1 Policy Usage and Conformance 35
5.7.2 Policy Description and Motivation 35
5.7.3 Policy Constraints 35
5.7.4 Policy 35
5.7.5 Policy Example(s) 36
5.7.6 Policy History 36
5.8 Policy on Logical Data Model (LDM) 36
5.8.1 Policy Usage and Conformance 36
5.8.2 Policy Description and Motivation 36
5.8.3 Policy Constraints 36
5.8.4 Policy 36
5.8.5 Policy Example(s) 37
5.8.6 Policy History 38
5.9 Policy on Logical Message Model 38
5.9.1 Policy Description and Motivation 38
5.9.2 Policy Constraints 38
5.9.3 Policy 39
5.9.4 Policy Example(s) 39
5.9.5 Policy History 39
5.10 Policy of Logical Service Model 39
5.10.1 Policy Usage and Conformance 39
5.10.2 Policy Description and Motivation 39
5.10.3 Policy Constraints 39
5.10.4 Policy 39
5.10.5 Policy Example(s) 39
5.10.6 Policy History 39

Documents
Document Sharing Hub
Full document details: SAE AS6522B

6. NON-FUNCTIONAL PROPERTY POLICIES 39
6.1 Policy on Non-Functional Property Library 39
6.1.1 Policy Usage and Conformance 39
6.1.2 Policy Description and Motivation 39
6.1.3 Policy Constraints 39
6.1.4 Policy 40
6.1.5 Policy Example(s) 40
6.1.6 Policy History 41

7. SYSTEM USE CASE MODEL POLICIES 41
7.1 Policy on Use Case Diagrams 41
7.1.1 Policy Usage and Conformance 41
7.1.2 Policy Description and Motivation 41
7.1.3 Policy Constraints 41
7.1.4 Policy 42
7.1.5 Policy Example(s) 43
7.1.6 Policy History 44
7.2 Policy on UML Activity Diagrams 44
7.2.1 Policy Usage and Conformance 44
7.2.2 Policy Description and Motivation 44
7.2.3 Policy Constraints 44
7.2.4 Policy 44
7.2.5 Policy Example(s) 45
7.2.6 Policy History 46
7.3 Policy on UML Sequence Diagrams 46
7.3.1 Policy Usage and Conformance 46
7.3.2 Policy Description and Motivation 46
7.3.3 Policy Constraints 46
7.3.4 Policy 46
7.3.5 Policy Example(s) 47
7.3.6 Policy History 48
7.4 Policy on SOA States and Modes Diagrams 48
7.4.1 Policy Usage and Conformance 48
7.4.2 Policy Description and Motivation 49
7.4.3 Policy Constraints 49
7.4.4 Policy 49
7.4.5 Policy Example(s) 49
7.4.6 Policy History 50
7.5 Policy on Services Architecture Diagrams 50
7.5.1 Policy Usage and Conformance 50
7.5.2 Policy Description and Motivation 50
7.5.3 Constraints 50
7.5.4 Policy 50
7.5.5 Policy Example(s) 51
7.5.6 Policy History 52

8. NOTES 52
8.1 Revision Indicator 52

Figure 1 UCS Architecture Model 10
Figure 2 UCS Architectural Model Packages 10
Figure 3 UCS Architecture Framework 11
Figure 4 Imported DMF Package structure 13
Figure 5 Example DataTypes that conform to the DMF and AS6969 14
Figure 6 Conceptual and example logical DataType refinements for mass 16
Figure 7 WGS84_EllipsoidPosition[rad,m] 16
Figure 8 Example logical DataTypes for radiant flux 17
Figure 9 Conceptual Model Packages 18
Figure 10 Example Conceptual Entity 18

AVA Documents
Document Sharing Hub
Full document details: SAE AS6522B

Figure 11 VehicleStatusType:: propulsionFuelLoad: Mass 19
Figure 12 VehicleFlightStatus ServiceInterface 20
Figure 13 Packages in the Common Package of Conceptual Entity Types 21
Figure 14 UCS Stereotypes Used in Conceptual Entity Types 21
Figure 15 DomainResourceType and Principal Parts of UCS System 22
Figure 16 DataProductsType and its Principal Specializations 22
Figure 17 UCSObjectiveType and ActionType 23
Figure 17 UCSMissionType 24
Figure 19 Packages in Conceptual Data Types Package 24
Figure 20 Air Sortie Mission Associations 25
Figure 21 Waypoint Mission Associations 26
Figure 22 Packages in Conceptual Message Model Package 27
Figure 23 UCSMissionStatusType Projection 28
Figure 24 SubsystemStatusReportType Projections 29
Figure 25 VehicleOperatingStateType Projections 30
Figure 26 Refinement of LandingGearStatusType 31
Figure 27 UCS SoaML UML profile 32
Figure 28 SOA participant diagram example 33
Figure 29 Protocol Activity diagram example 1 34
Figure 30 Protocol Activity diagram example 2 34
Figure 31 Logical Model Packages 35
Figure 32 Conceptual and Logical UCSSystemType 38
Figure 33 Context of TopLevelMishap Package 40
Figure 34 TopLevelMishapType 41
Figure 35 Perform UAS Operations Use Case 42
Figure 36 Use case diagram example 1 43
Figure 37 Use case diagram example 2 43
Figure 38 Activity diagram example 1 45
Figure 39 Activity diagram example 2 46
Figure 40 UML sequence diagram example 1 47
Figure 41 UML sequence diagram example 2 48
Figure 42 System states and modes diagram 49
Figure 43 L3 ServicesArchitecture diagram example 51
Figure 44 L2 ServicesArchitecture diagram example 52

Table 1 Projection connector guidelines 27

Documents
Document Sharing Hub

Full document details: SAE AS6522B

1. SCOPE

This Technical Governance is part of the SAE UCS Architecture Library and is primarily concerned with the UCS Architecture Model (AS6518) starting at Revision A and its user extensions. Users of the Model may extend it in accordance with AS6513 to meet the needs of their UCS Products. UCS Products include software components, software configurations and systems that provide or consume UCS services. For further information, refer to AS6513 Revision A or later.

Technical Governance is part of the UCS Architecture Framework. This framework governs the UCS views expressed as Packages and Diagrams in the UCS Architecture Model.

2. TERMINOLOGY

In this document, all strict textual references to Unified Modeling Language (UML) Metaclasses and Stereotypes will begin with a capital letter and follow the naming conventions in the applicable standard. All strict textual references to instances of Metaclasses and Stereotypes (including notional examples) will be indicated in bold text.

For example, the UCS Architecture Model comprises the **Notices** Package, **Data Model Framework** Package, **UCS Architectural Model Package** and **UCS Architecture Technical Goverance** Package. The Stereotype **ScalarQuantity** Extends DataType. An Instance of this Stereotype is **Mass[kg]** whose Attribute **number** denotes the magnitude of mass in kilograms.

3. REFERENCES

3.1 Applicable Documents

The following publications form a part of this document to the extent specified herein. The latest issue of SAE publications shall apply. The applicable issue of other publications shall be the issue in effect on the date of the purchase order. In the event of conflict between the text of this document and references cited herein, the text of this document takes precedence. Nothing in this document, however, supersedes applicable laws and regulations unless a specific examption has been obtained.

3.1.1 SAE Publications

Available from SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001, Tel: 877-606-7323 (inside USA and Canada) or +1 724-776-4970 (outside USA), www.sae.org.

AS6512A	SAE Unmanned Systems (UxS) Control Segment (UCS) Architecture: Architecture Description
AS6513A	SAE Unmanned Systems (UxS) Control Segment (UCS) Architecture: Conformance Specification
AS6518A	SAE Unmanned Systems (UxS) Control Segment (UCS) Architecture: Architecture Model
AS6969A	SAE Data Dictionary for Quantities Used in Cyber Physical Systems

3.1.2 Object Management Group (OMG) Publications

OMG OMG Unified Modeling Language® (UML®). Version 2.5.1. December 2011. Available online at: https://www.omg.org/spec/UML/2.5.1.

OMG Service Oriented Architecture Modeling Language (SoaML). Version 1.0.1. May 2012. Available online at: https://www.omg.org/spec/SoaML/1.0.1.

OMG System Modeling Language (SysML®). Version 1.6. December 2019. Available online at: https://www.omg.org/spec/SysML/1.6.

Documents
Document Sharing Hub
Full document details: SAE AS6522B

3.1.3 ISO Publications

Copies of these documents are available online at http://webstore.ansi.org/.

ISO/IEC 80000 Quantities and Units

3.1.4 ECMA International Publications

Standard EMCA-262 ECMAScript® 2019 Language Specification. 10th edition (June 2019).

3.1.5 U.S. Government Publications

Available from U.S. Army Combat Capabilities Development Command (CCDC), Ground Vehicle Systems Center, 6305 E 11 Mile Rd, Warren, MI 48092.

U.S Government. Data Model Framework (DMF). Version 1.1. April 2020.

3.2 Definitions

This technical governance conforms to the definitions provided in the Data Model Framework and OMG standards cited in 3.1.

3.3 Acronyms

CDM Conceptual Data Model

DMF Data Model Framework

GUID Global Unique Identifier

ISO International Organization for Standardization

LDM Logical Data Model

NFP Non-Functional Property

OMG Object Management Group

QoS Quality of Service

SOA Service Oriented Architecture

SoaML Service Oriented Architecture Modeling Language

UAF Unified Architecture Framework

UCS UxS Control Segment

UML Unified Modeling Language

UxS Unmanned System

XML Extensible Markup Language

Documents
Document Sharing Hub

Full document details: SAE AS6522B

4. GENERAL POLICIES

4.1 Policy on Namespace

4.1.1 Policy Usage and Conformance

This policy applies to the UCS Architecture Model and any derived SAE-managed artifacts delivered in XML format.

4.1.2 Policy Description and Motivation

This policy defines the UCS convention for declaring the Namespace of XML Artifacts. The policy is intended to avoid clashes between names from different markup vocabularies.

4.1.3 Policy Constraints

The definition Objects defined by SAE AS6969 (beginning in Revision A) use the Namespace name **https://www.sae.org/AS6969/**.

4.1.4 Policy

All UCS XML architecture products will declare the UCS XML Namespace using the UCS Namespace identifier **http**s**://www.**sae**.org**/AS6512/ where AS6512 is the designation for the UCS Architecture Library, including the UCS Architecture Model. This is represented as an xmlns XML attribute:

xmlns: ucs="https://www.sae.org/AS6512/"

In addition to the Namespace declaration, each release of the UCS Architecture XML products must contain the release identifier as a local name using a version attribute. For example, for Revision A:

xmlns: ucs="https://www.sae.org/AS6512/" version="Revision A"

4.1.5 Policy Example(s)

None.

4.1.6 Policy history

Updated in Revision A from AS6522 UCS-TECHGOV-POLICY Namespace.

4.2 Policy on Model Languages

4.2.1 Policy Usage and Conformance

This policy applies to the UCS Architecture Model, AS6518, starting at Revision A. The model will conform to this policy.

4.2.2 Policy Description and Motivation

This policy identifies the Unified Modeling Language (UML) specifications and industry Profiles of UML to be Applied in the UCS Architcture Model.

4.2.3 Policy Constraints

Contraints on this policy are established by other policies in this technical governance.

Documents
Document Sharing Hub
Full document details: SAE AS6522B

4.2.4 Policy

The UCS Architecture Model shall be expressed in UML 2.5.1 sing industry profiles as determined by the subject matter. These industry profiles are:

1. OMG Service Oriented Architecture Modeling Language (SoaML) version 1.0.1

2. OMG System Modeling Langauge (SysML) version 1.6

3. OMG Unified Archirtecture Framework (UAF) Profile (UAFP) version 1.0

4. U.S. Government profiles defined by the Data Model Framework (DMF) version 1.1

In addition, the UCS Architecture Model defines profiles of UML as established in policies in this technical governance. The UCS Profile Diagrams are provided in the **UCS Architecture UML Profiles** Package of the **UCS Architecture Technical Governance** Package.

4.2.5 Policy Examples

None.

4.2.6 Policy History

Original at Revision A.

4.3 Policy on UCS Architecture Model Package Structure

4.3.1 Policy Usage and Conformance

The policy applies to the UCS Architecture Model, AS6518, beginning at Revision A. The model will conform to this policy.

4.3.2 Policy Description and Motivation

This policy describes the high level Package structure of the UCS Architecture Model. Further policies will expand upon these Packages.

4.3.3 Policy Constraints

None.

4.3.4 Policy

The UCS Architecture Model comprises four model views as shown in Figure 1. These model views are the **UCS Architectural Model** Package, the **UCS Architecture Technical Goverance** Package, the **Data Model Framework** Package, and **NOTICES** Package. It will be noted in this terminology that the **UCS Architectural Model** is a Package within the UCS Architecture Model.

Documents
Document Sharing Hub
Full document details: SAE AS6522B

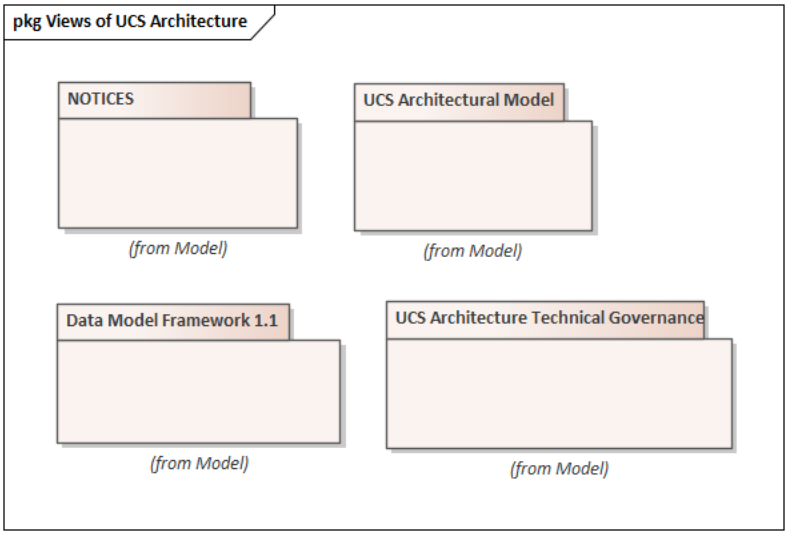

Figure 1 - UCS Architecture Model

The **Notices** Package contains publication metadata about the UCS Architecture Model.

The **UCS Architectural Model** is a Package that contains the UCS Architecture information model at two levels of abstraction called the **Conceptual Model** and the **Logical Model**. These abstractions are based on the abstraction of their DataTypes as defined in the **Data Model Framework 1.1** Package (DMF). The DMF includes the definitions provided in AS6969. The **Conceptual Model** and **Logical Model** have a related but not identical Package structure. Each contains a service model, a message model, and a datamodel. Policies concerning these models are provided in Section 5.

Supporting the information model is the **Non-Functional Property Model** Package, the **System Use Case Model** Package, and the **Model Overview** Package. These Packages are provided for reference only and are not normative. Policies concerning the **Non-Functional Property Model** are provided in Section 6. Policies concerning the **System Use Case Model** are provided in Section 7. See Figure 2.

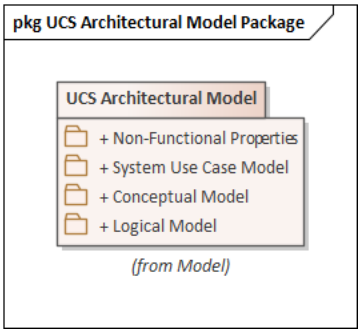

Figure 2 - UCS Architectural Model Packages

The architecture framework for the **UCS Architectural Model** is provided by the **UCS Architecture Technical Governance Package** and **Data Model Framework 1.1** Package. The **UCS Architecture Technical Governance** Package contains the UCS Architecture Profiles of UML together with this document. The **DMF** Package is a Government-owned plug-in to the UCS Architecture Model. It contains Profiles of UML, Technical Governance, and the AS6969 Library of definition Objects upon which the DMF is based. The **DataTypes: Common** Package in the DMF contains an example library of conceptual and logical DataTypes. These DataTypes at the Global Unique Identifier (GUID) level are not used in the **UCS Architectural Model** Package. The **UCS Architectural Model** Package contains its own DataTypes (with unique GUIDs).

Documents
Document Sharing Hub
Full document details: SAE AS6522B

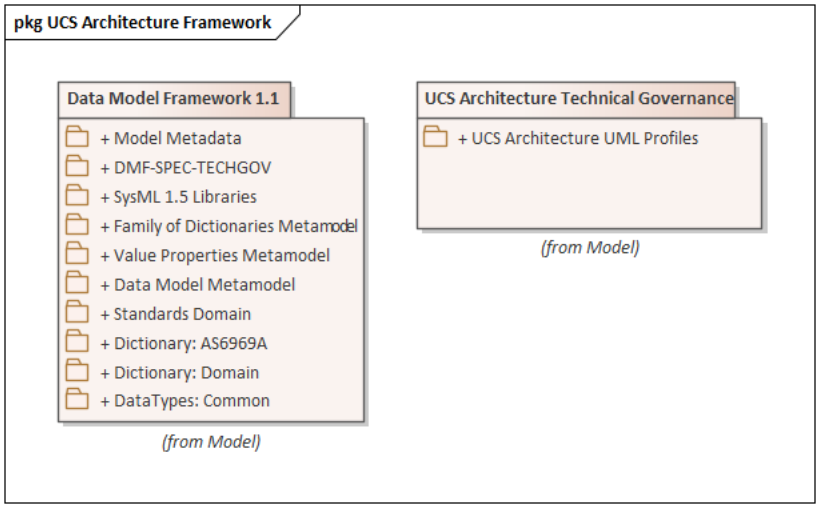

Figure 3 - UCS Architecture Framework

4.3.5 Policy Example(s)

None.

4.3.6 Policy History

Updated in Revision A from AS6522 UCS-TECHGOV-POLICY Architecture Package Structure.

4.4 General Policy on Naming Conventions

4.4.1 Policy Usage and Conformance

This policy applies to NamedElements in the UCS Architecture Model. It should be noted that UML-based modeling languages, including SysML and UAF, do not formally impose naming conventions.

4.4.2 Policy Description and Motivation

This policy establishes general model naming conventions. These conventions will promote a uniform modeling style and make it easier to navigate and understand the UCS Architecture Model and associated work products that reference those models. Additional naming policies are provided for specific Packages in the UCS Architecture Model.

4.4.3 Policy Constraints

This policy does not apply to separately governed third-party content that has been imported into UCS Architecture Model.

4.4.4 Policy

NamedElements in the model shall have names that satisfy the regular expression specified below for each category of name. The regular expressions specified below satisfy JavaScript conventions (ECMA-262) and are used verbatim in name checks.

Package and Type name conventions permit an underscore "_" prefix during model development that indicates that an element is work-in-progress and not part of the formal model.

Documents
Document Sharing Hub
Full document details: SAE AS6522B

4.4.4.1 Package and Use Case Names

Allow spaces, periods, hyphens, parentheses.

```
var legalPackageOrUseCaseName = new RegExp("^_?[A-Z][a-zA-Z0-9_.() -]*$");
```

4.4.4.2 General Type Names (Classes, Enumerations, Components, Actors)

Type names follow the general "camel case" convention of beginning with an uppercase letter.

```
var legalTypeName = new RegExp("^_?[A-Z][a-zA-Z0-9_-]*$");
```

Service Interfaces allow a single preceding "~" character.

```
var legalTypeNameConjugate = new RegExp("^~[A-Z][a-zA-Z0-9_]*$");
```

4.4.4.3 General Instance Names (Attributes, Operations)

Instance names follow the general "camel case" convention of beginning with a lowercase letter.

```
var legalInstanceName = new RegExp("^[a-z][A-Za-z0-9_]*$");
```

Instance names also permit an initial acronym where an acronym consists of at least two uppercase letters, or an uppercase letter and a digit.

```
var legalInstanceNameAcroPrefix = new RegExp("^[A-Z][A-Z0-9][A-Za-z0-9_]*$");
```

Enumeration literals follow the convention of all uppercase letters with digits and underscores.

```
var legalEnumerationAttributeName = new RegExp("^[A-Z][A-Z0-9_]*$");
```

4.4.4.4 Specific Instance Names (Attributes, Operations)

Time stamps:

Except as allowed by this set of rules, all Attributes relating to time or time stamps shall use the term "**timeStamp**." The term **timeStamp** shall be considered as the concatenation of the two words "time" and "stamp." This distinction is specified for camel case guidelines.

The Attribute name may include an additional identifier, but the name shall still use the term **timeStamp** with the appropriate camel case (e.g., **previousTimeStamp**). If the Attribute belongs to an Element realizing a NamedElement in the **Conceptual Model** that already has a time-related Attribute (e.g**., TimeCapabilityType**), the Attribute name shall use the name present in the **Conceptual Model** Element (e.g., **timeSetPoint** would be named **timeSetPoint** and not **timeStampSetPoint**).

4.4.5 Policy Example(s)

See the policy examples throughout this document.

4.4.6 Policy History

Updated in Revision A from AS6522 UCS-TECHGOV-POLICY Naming Conventions. Naming conventions for content in specific Packages is now provided in the appropriate policy.

Documents
Document Sharing Hub
Full document details: SAE AS6522B

5. INFORMATION MODEL POLICIES

These policies apply to the **Conceptual Model** Package and **Logical Model** Package, which provide the core views of the UCS Architecture Model at different levels of abstraction**.** The defining difference between these models is the level of abstraction of their DataTypes. Both models contain a service model, a message model, and a datamodel.

The principal Package in the **Conceptual Model** is the **Conceptual Data Model** (CDM). This is a definitional model of the entities in the UCS Domain. In this definitional model, the DataTypes serve only to scope the semantics of an entity's innate Properties. The main purpose of the **Conceptual Message Model** and **Conceptual Service Model** is to show the trace from the **Logical Message Model** to the CDM. The conformance of UCS Products to the UCS Architecture is based on the **Logical Service Model** whose Interface Signals are parameterized by logical Messages defined in the **Logical Message Model**. In the **Logical Message Model**, DataTypes specify the Value structure so that Values can be interpreted.

For example, the conceptual DataType **Mass** would be used only to Type a particular Property as a mass property (e.g., **Aircraft:: fuelAmount: Mass**). The Features of this Datatype are insufficient to interpret a Value of **Mass**. The logical DataType **Mass[kg]** is a refinement of **Mass**. Its Value is **number: Real** based on the arbitrarily chosen kilogram measurement unit.

5.1 Policy on Conformance to the Data Model Framework

5.1.1 Policy Usage and Conformance

This policy applies to the UCS Architecture Model, most specifically the **Conceptual Data Model** Package and **the Logical Data Model** Package. The UCS Architecture Model shall conform to this policy starting at Revision A.

5.1.2 Policy Description and Motivation

The **Data Model Framework 1.1** is a Package in the UCS Architecture Model.

The Data Model Framework (DMF) is a government owned and managed architecture framework for datamodels. It is founded on the mathematical principles that were established in the SAE AS6969 document, Data Dictionary for Quantities Used in Cyber Physical Systems. The Profiles of UML provided by the DMF are the **FOD** Profile, **DataType** Profile, **Value Property** Profile, and **Data Model Packages** Profile. The SAE AS6969 Model, starting at Revision A, is based on the DMF **FOD Profile**.

The Package structure for the imported elements of the Data Model Framework is shown in Figure 4. It includes the Package **Dictionary: AS6969A**, which is the definition library from AS6969 starting at Revision A. Adoption of the DMF provides the opportunity for a consistent approach to data dictionaries and datamodels across multiple government-sponsored architectures.

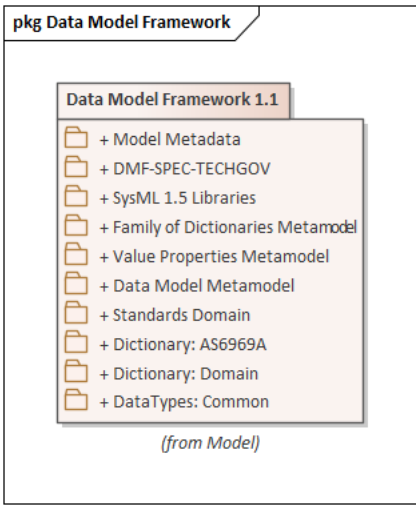

Figure 4 - Imported DMF Package structure

Documents
Document Sharing Hub
Full document details: SAE AS6522B

5.1.3 Constraints

The UCS Architectue Model is in the process of transitioning to towards full alignment with the DMF. For Revision A of the UCS Architecture Model, conformance does not include nominal properties, property modals or Enumerations. Not all DataTypes fully align with DMF in Revision A.

5.1.4 Policy

The UCS Architecture Model shall conform to the polices and UML profiles defined in the **Data Model Framework** Package. The document Artifact **DMF-SPEC-TECHGOV** contained with the **Data Model Framework** Package is a part of this technical governance by reference.

The **Data Model Framework** package, created through collaboration with the U.S. Army Autonomous Ground Vehicle Reference Architecture (AGVRA) effort, has been adopted and imported wholly into the UCS Architecture Model. Its models will be referenced and extended as needed by UCS, but the DMF package contents will not be edited by UCS.

The DMF Package **DataTypes: Common** provides example DataTypes. This Package is not directly used in the **UCS Architectural Model** Package. The **UCS Architectural Model** Package contains its own library of DataTypes. Many of these differ from their DMF equivalent only in the DataType's GUID.

5.1.5 Policy Example(s)

Begin Example

Figure 5 is from an example in Policy 7.03 of DMF-SPEC-TECHGOV. It shows the conceptual DataType **ThermodynamicTemperature** and the derived logical DataType **ThermometerTemperature[C]**. The semantics of these DataTypes is supported by definitions provided by AS6969. These are the definitions of **Thermodynamic Temperature Quantity**, **Celsius Temperature Quantity**, and **Degree Celsius**.

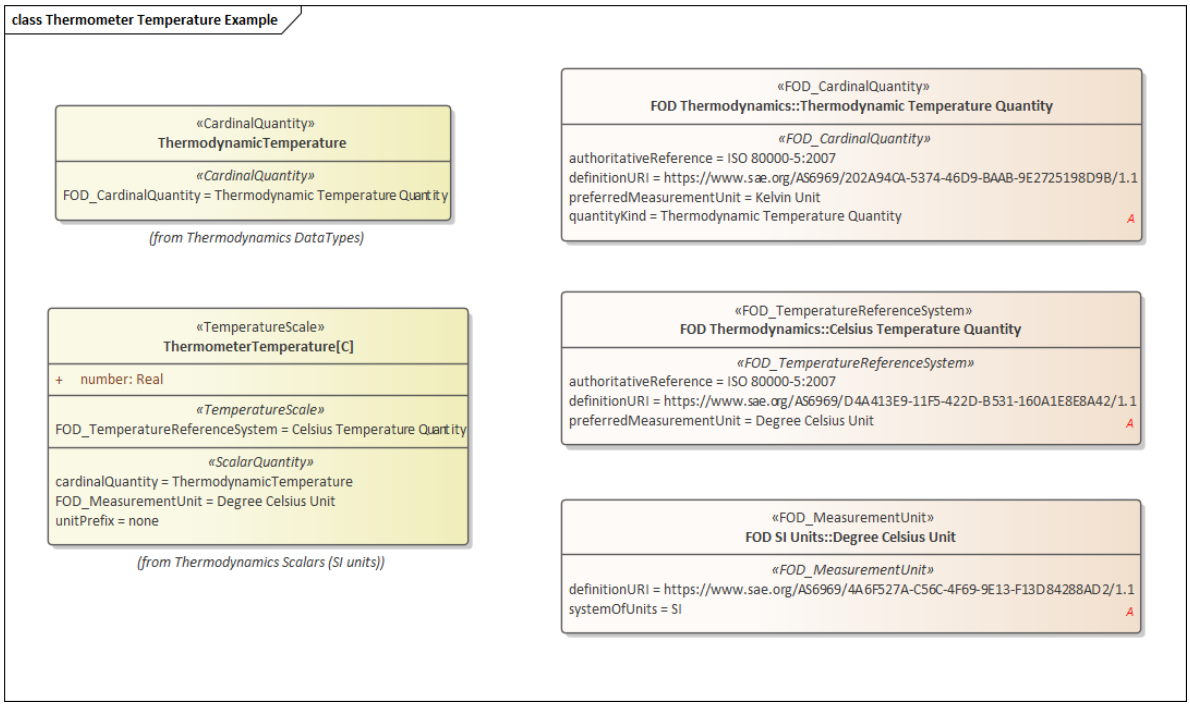

Figure 5 - Example DataTypes that conform to the DMF and AS6969

End Example

Documents
Document Sharing Hub
Full document details: SAE AS6522B

5.1.6 Policy History

New at Revision A.

5.2 Policy on Levels of Architecture Abstraction

5.2.1 Policy Usage and Conformance

This policy applies to the **Conceptual Data Model** Package and **Logical Data Model** Package within the **UCS Architectural Model** Package. These Packages shall conform to this policy.

5.2.2 Policy Description and Motivation

The defining difference between the **Conceptual Data Model** and **Logical Data Model** is the level of abstraction of the Datatypes used in each model. Therefore a clear policy is required to distiguish between a conceptual DataType and a logical DataType. The distinction is based on DMF Technical Goverance Policy 8.01 (DataType Abstractions). The DMF is based on AS6969, which defines basic quantities, their modalities, and their morphisms. The morphism of a quantity is considered a logical specification.

It shall be noted that a conceptual specification is not intended to be vague. Force is precisely and completey defined in ISO 80000-4 as the rate of change of momentum. The logical value of a force vector may have many morphisms based on different vector spaces, measurement units, and unit prefixes. For example, the components of force may use kN in a North-East-Down vector space. The same value might also be expressed in pounds-force in an East-North-Up vector space.

5.2.3 Constraints

None.

5.2.4 Policy

The **Conceptual Model** shall use DataTypes as defined in DMF Technical Governance Policy 8.04 (Conceptual DataTypes) and DMF Technical Goverance Policy 8.05 (Conceptual Enumerations). DataTypes used in the **Conceptual Model** shall apply the Stereotypes defined in the DMF **DataType Profile** Package.

The **Logical Model** shall use DataTypes as defined in DMF Technical Governance Policy 8.07 (Logical DataTypes) and DMF Technical Goverance Policy 8.08 (Logical Enumerations). DataTypes used in the **Logical Model** shall apply the Stereotypes defined in the DMF **DataType Profile** Package.

It will be noted that logical DataTypes for basic quantities have a **number** Attribute whose Type is the SysML Primitive **Number** or its Specializations **Integer**, **Real** or **Complex**. The SysML **Number**, **Real** and **Complex** Primitives are to be interpreted as continuous variables whose range is only limited by the semantics of the quantity magnitude they express. It is only at the software implementation level that the Value Primitive is to be interpreted as a discrete variable that would occupy finite memory.

5.2.5 Policy Example(s)

Begin Example

Figure 6 is based on DMF-SPEC-TECHGOV Policy 8.01 (DataType Abstractions). It shows a conceptual DataType called **Mass** and two logical refinements of **Mass** called **Mass[kg]** and **Mass[mg].** The conceptual DataType is a Primitive with no Value Attribute. Its purpose is only to scope the semantics of a Property (measurand) in a conceptual data model. For example, the measurand **Aircraft:: fuelAmount: Mass** has a different meaning to **Aircraft:: fuelAmount: Volume**. A conceptual data model is a definitional model of entities and their innate properties. The definition of **Aircraft:: fuelAmount: Mass** does not depend on the chosen measurement unit.

In contrast to a conceptual data model, a logical data model is concerned with how the state of an entity is to be expressed. The two logical DataTypes in the figure provide alternate value structures for **Mass**. Consider the value 0.001 kg = 1000 mg. In the DataType **Mass [kg]**, the **number: Real** part of the value is 0.001. In the DataType **Mass [mg]**, the **number: Real** part of the value is 1000. These two DataTypes therefore provide different morphisms of the same value.

AVA Documents
Document Sharing Hub