

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Asset Administration Shell for industrial applications –
Part 1: Asset Administration Shell structure**

**Enveloppe de Gestion d'Actif pour applications industrielles –
Partie 1: Structure de l'Enveloppe de Gestion d'Actif**

IECNORM.COM : Click to view the full PDF of IEC 63278-1:2023



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2023 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Secretariat
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

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

About the IEC

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

About IEC publications

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

IEC publications search - webstore.iec.ch/advsearchform

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 It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

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

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch

Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org

The world's leading online dictionary on electrotechnology, containing more than 22 300 terminological entries in English and French, with equivalent terms in 19 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Recherche de publications IEC -

webstore.iec.ch/advsearchform

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études, ...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et une fois par mois par email.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch

Découvrez notre puissant moteur de recherche et consultez gratuitement tous les aperçus des publications. Avec un abonnement, vous aurez toujours accès à un contenu à jour adapté à vos besoins.

Electropedia - www.electropedia.org

Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 300 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 19 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Asset Administration Shell for industrial applications –
Part 1: Asset Administration Shell structure**

**Enveloppe de Gestion d'Actif pour applications industrielles –
Partie 1: Structure de l'Enveloppe de Gestion d'Actif**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 71.100.20

ISBN 978-2-8322-7679-2

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	5
INTRODUCTION.....	7
0.1 General.....	7
0.2 Overview on parts of the series.....	7
0.3 Interoperability.....	8
0.4 Key objectives of the Asset Administration Shell	9
1 Scope.....	10
2 Normative references	10
3 Terms, definitions, abbreviated terms, and conventions.....	10
3.1 Terms and definitions.....	10
3.2 Abbreviated terms.....	15
3.3 Conventions.....	17
4 Conceptual framework	17
4.1 General.....	17
4.2 Asset Administration Shell and related entities.....	17
4.2.1 General	17
4.2.2 Detailed overview	18
4.2.3 Asset	19
4.2.4 AAS responsible	20
4.2.5 AAS user application	20
4.2.6 Asset Administration Shell	20
4.2.7 AAS interface	21
4.2.8 Submodel	21
4.2.9 SubmodelElement.....	22
4.2.10 Submodel template.....	22
4.2.11 Submodel template element.....	22
4.2.12 Concept repositories.....	23
4.2.13 Asset integration.....	25
4.2.14 Asset service	25
4.2.15 Asset related services	25
4.3 Life cycle aspects of assets and Asset Administration Shells	26
4.4 Example for an overall Asset Administration Shell scenario	28
5 Identifiers	31
5.1 Needs	31
5.2 Determination of identifiers	31
5.2.1 General	31
5.2.2 Globally distinct identifiers for concepts by IRDIs.....	32
5.2.3 Globally distinct identifiers by URIs.....	32
5.2.4 Local identifiers	32
6 Asset Administration Shell structure	33
6.1 General.....	33
6.2 Requirements associated to Asset Administration Shell	34
6.2.1 General	34
6.2.2 Asset Administration Shell	34
6.2.3 Submodel	35
6.2.4 SubmodelElements.....	36

6.2.5	AAS interface	40
6.3	Requirements related to data exposure and information security.....	41
6.3.1	Data exposure of Asset Administration Shell and its Submodels	41
6.3.2	Requirements related to information security of Asset Administration Shell in general	42
6.3.3	Requirements related to the IEC 62443 series	43
Annex A (informative)	Relevant standards	44
A.1	Possible sources for Submodels and Submodel templates	44
A.1.1	General	44
A.1.2	Different sets of concept definitions for SubmodelElements	44
A.1.3	Existing international specifications that can be used as basis for Submodel templates	45
A.2	IEC 61360 dictionaries, classes and property types	48
A.2.1	General	48
A.2.2	Classes	49
A.2.3	Property types and instances.....	49
A.2.4	IEC Common Data Dictionary (IEC CDD).....	50
A.2.5	ECLASS	51
A.3	IEC 61987 series classes and dictionary	51
A.3.1	General	51
A.3.2	Specific classes	52
A.3.3	Dictionary	53
A.4	IEC 62683 series classes and dictionary	53
A.4.1	General	53
A.4.2	Dictionary	54
A.5	Digital Factory (IEC 62832 series)	54
A.5.1	Introduction to Digital Factory	54
A.5.2	Compatibility of the Digital Factory with the concept of Asset Administration Shell	57
A.6	AutomationML (IEC 62714 series).....	59
A.6.1	AutomationML overview.....	59
A.6.2	AutomationML modeling concepts.....	61
A.6.3	Interoperability of Asset Administration Shell supported by AutomationML	62
A.7	OPC UA	63
A.7.1	OPC UA overview	63
A.7.2	OPC UA Information Models	65
A.7.3	Relationship between AutomationML and OPC UA	66
Annex B (informative)	Usage view of the Asset Administration Shell	67
Annex C (informative)	Security for industrial automation and control systems	69
C.1	Security concepts from the IEC 62443 series	69
C.2	Foundational requirements.....	69
C.3	Security level	70
C.4	Measures of defence for IACS	70
Bibliography		71
Figure 1 – Facets of interoperability according to ISO/IEC 21823-1		8
Figure 2 – Overview of Asset Administration Shell and related entities.....		17
Figure 3 – Information exchange between AAS user applications.....		18

Figure 4 – Detailed overview of Asset Administration Shell and related entities	18
Figure 5 – Assets seen as type asset or instance asset	19
Figure 6 – Example of different Asset Administration Shells associated to the same asset.....	21
Figure 7 – Example of different kinds of concept repositories referenced by SubmodelElements	23
Figure 8 – Top level concepts and relationships of "Capability for Industry Ontology"	24
Figure 9 – Example of modelling by means of Asset Administration Shell	25
Figure 10 – Example of asset integration, asset services and asset related services	25
Figure 11 – Example of life cycle aspects of assets and Asset Administration Shells	26
Figure 12 – Example of assets in product and production system life cycles	27
Figure 13 – Example of Asset Administration Shell in product life cycle	28
Figure 14 – Illustration of an example of an overall Asset Administration Shell scenario	28
Figure 15 – Illustration of value exchange in overall scenario.....	30
Figure 16 – Different identifiers for globally distinct identifiers and local identifiers	32
Figure 17 – Example of an Asset Administration Shell demonstrating the general structure	33
Figure 18 – Illustration of different aspects of SubmodelElements.....	36
Figure 19 – Asset Administration Shells of a representative assembly of electrical axes	39
Figure 20 – AAS user application accessing AAS interfaces of Asset Administration Shells	41
Figure 21 – Asset Administration Shell security overview.....	42
Figure A.1 – Example: Representation of a class tree in IEC CDD	49
Figure A.2 – Simplified UML scheme of device, LOPs and aspects (see IEC 61987-11).....	52
Figure A.3 – Organization in blocks	54
Figure A.4 – Structured asset class in the Digital Factory (see IEC 62832 series).....	55
Figure A.5 – Description of a structured asset that is composed of several assets	56
Figure A.6 – Comparison of the approaches of the Digital Factory and Asset Administration Shell	57
Figure A.7 – Architecture of AutomationML	60
Figure A.8 – CAEX concepts.....	61
Figure A.9 – Relations between AutomationML elements	62
Figure A.10 – OPC UA specification organization.....	64
Figure A.11 – Component internal base mode.....	65
Figure B.1 – Overview of usage view of the Asset Administration Shell [7].....	67
Figure C.1 – Foundational Requirements and Security Levels applicable for an Asset	70
Table 1 – Examples for categorization of SubmodelElements	38
Table A.1 – Examples of standards providing concept repository entries which can be referenced by Submodel templates	46
Table A.2 – Examples of standards providing potential sources of Submodel templates.....	47
Table A.3 – Examples of standards providing reference models for Submodels	48
Table A.4 – Example of representation of a property type with some attributes in IEC CDD.....	50
Table A.5 – Comparison of the individual concepts of Digital Factory (IEC 62832 series) and Asset Administration Shell	58

INTERNATIONAL ELECTROTECHNICAL COMMISSION

ASSET ADMINISTRATION SHELL FOR INDUSTRIAL APPLICATIONS –**Part 1: Asset Administration Shell structure**

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. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements 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.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC 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 transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is 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 its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees 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 Publication or any other IEC 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 some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

IEC 63278-1 has been prepared by IEC technical committee 65: Industrial-process measurement, control and automation. It is an International Standard.

The text of this International Standard is based on the following documents:

Draft	Report on voting
65/1012/FDIS	65/1027/RVD

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

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/publications.

A list of all parts in the IEC 63278 series, published under the general title *Asset Administration Shell for industrial applications*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

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

IMPORTANT – The "colour inside" logo on the cover page of this document indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

IECNORM.COM : Click to view the full PDF of IEC 63278-1:2023

INTRODUCTION

0.1 General

The production system life cycle focuses on the design, deployment, commissioning, operation and decommissioning of an entire production facility. Product life cycle management is the process of managing the entire life cycle of a product with the information flows and controls from inception, through engineering design and manufacture, to service and end of life treatment of manufactured products. The supply chain management is the management of the flow of products and services and includes processes that transform raw materials and parts components into final products, and it involves the streamlining of business activities to maximize customer value and gain a competitive advantage in the marketplace. Each of these dimensions intersects at the vertical integration of machines, plants, and enterprise systems in the equipment hierarchy of an enterprise pyramid. The integration of manufacturing software applications along each dimension and across dimensions helps to enable advanced controls at the shop floor and optimal decision-making at the enterprise. Details of existing manufacturing standards for each of the three life cycle dimensions are provided in [1]¹.

Several integration technologies have been individually put into practical use (e.g. CAD/CAM) aiming to accelerate product innovation cycles, streamline supply chains, and increase production system flexibility through information exchange between the dimensions. Details of the integration technologies and capabilities supported by them are provided in [1].

The Asset Administration Shell (AAS) is seen as one interoperable manifestation of a digital twin in manufacturing that facilitates tighter integration within and across the three dimensions mentioned above.

This document is the first part of the series "Asset Administration Shell for industrial applications". The multiple parts of the series will detail structure, information models, definition of services and online interfaces, required security aspects and communication languages including mapping contents of OPC UA and AutomationML models to the Asset Administration Shell.

0.2 Overview on parts of the series

The current planning foresees parts covering the following topics:

- Asset Administration Shell structure (this document)
- information meta-model (to allow to access standardized information)
- security provisions for Asset Administration Shells
- use cases and modelling examples
- interfaces to Asset Administration Shells
- communication language among sets of Asset Administration Shells
- specification of content of Asset Administration Shells for various domains

This part of IEC 63278 describes requirements towards the general structure, that each possible Asset Administration Shell should comply with. In a following part of the series, this structure will be developed further towards a meta-model of the Asset Administration Shell. Based on these specifications, individual Asset Administration Shells can be created. These individual Asset Administration Shells will be the actual containers of information and will provide information and services with respect to the described asset.

¹ Numbers in square brackets refer to the Bibliography.

0.3 Interoperability

The Asset Administration Shell pursues the overall purpose to support interoperability of software applications. According to ISO/IEC 21823-1, different facets for interoperability can be considered (see Figure 1).

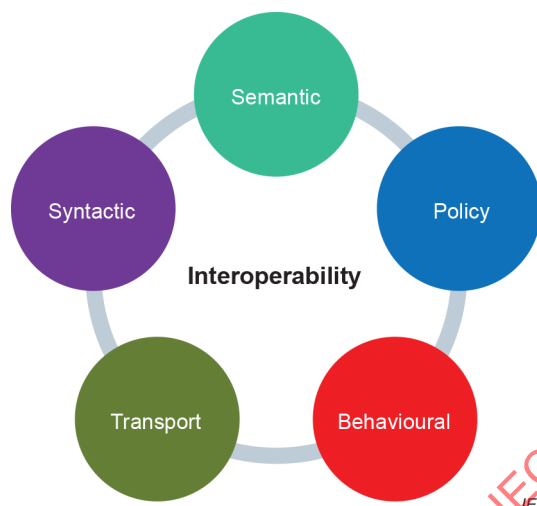


Figure 1 – Facets of interoperability according to ISO/IEC 21823-1

Semantic interoperability considers the meaning of the data model within the context of a subject area so that it is understood by the participating software applications. The Asset Administration Shell addresses semantic interoperability by associating well-known concepts to the data, which is exchanged between the software applications.

Policy interoperability considers the compliance with the legal, organizational, and policy frameworks applicable to the participating software systems. The Asset Administration Shell addresses policy interoperability in the following way:

- The Asset Administration Shell provides uniform identity and access control management including usage restriction for information and services of assets.
- The Asset Administration Shell enables uniform structuring of information and services of assets. This allows the Asset Administration Shell to define and maintain the structure of information and services of an asset and not the individual software applications. This simplifies information management in manufacturing industries by both reducing the effort and increasing the quality of information.

Transport interoperability considers the data transfer between software applications based on an established communication infrastructure between the participating software applications. This facet is not addressed in this part of the series but will be considered in further parts of the series.

Syntactic interoperability considers the data format by which the exchanged information can be understood by the participating software applications. This facet is not addressed in this part of the series but will be considered in further parts of the series.

Behavioural interoperability considers the expected outcomes to interface operations. This facet is addressed by the Asset Administration Shell in the sense that the Asset Administration Shell provides a standardized interface to software applications. The concrete behaviour of this standardized interface will be considered in further parts of the series.

0.4 Key objectives of the Asset Administration Shell

The following statements summarize these discussions and formulate some aims for the Asset Administration Shell, helping to keep the focus:

- Asset Administration Shell aims at establishing cross-company interoperability. Assets within manufacturing are provided by many different enterprises. In order to fulfil the scenarios of today and tomorrow, information and services on assets should be interoperable.
- Asset Administration Shell is intended for non-intelligent and intelligent products. The concept of asset comprises many different entities, with or without the ability to communicate actively or being intelligent. To leverage benefits in engineering, maintenance or operation throughout all hierarchy levels, the idea of the Asset Administration Shell is suitable to be applied by all assets.
- Asset Administration Shell aims at covering the complete life cycle of products, devices, machines and facilities.
Much useful information on assets is formed in the early phase of their life cycle, such as design, engineering and marketing. To maintain economic efficiency, digitized information from these early phases should be preserved and used in later phases, such as engineering higher level structures and operating and maintaining these structures.
- Asset Administration Shell aims at enabling integrated value chains.
Assets for manufacturing lines and products are provided by many different value chain partners. To maintain economic efficiency, digitized information should be exchanged among value chain partners. This will also enable advanced production modes (see 0.1).
- Asset Administration Shell is intended to be a base for autonomous systems and artificial intelligence.
In the future, many benefits are expected from approaches such as autonomous systems and artificial intelligence. These approaches require a sound basis of information and identifiers of elements. The Asset Administration Shell provides both.

ASSET ADMINISTRATION SHELL FOR INDUSTRIAL APPLICATIONS –

Part 1: Asset Administration Shell structure

1 Scope

This part of IEC 63278 defines the structure of a standardized digital representation of an asset, called Asset Administration Shell (AAS). The Asset Administration Shell gives uniform access to information and services.

The purpose of the Asset Administration Shell is to enable two or more software applications to exchange information and to mutually use the information that has been exchanged in a trusted and secure way.

This document focuses on Asset Administration Shells representing assets of manufacturing enterprises including products produced by those enterprises and the full hierarchy of industrial equipment. It defines the related structures, information, and services.

The Asset Administration Shell applies to:

- any type of industrial process (discrete manufacturing, continuous process, batch process, hybrid production);
- any industrial sector applying industrial-process measurement, control and automation;
- the entire life cycle of assets from idea to end of life treatment;
- assets which are physical, digital, or intangible entities.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 62443 (all parts), *Security for industrial automation and control systems*

3 Terms, definitions, abbreviated terms, and conventions

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- IEC Electropedia: available at <https://www.electropedia.org/>
- ISO Online browsing platform: available at <https://www.iso.org/obp>

3.1.1**asset**

entity owned by or under the custodial duties of an organization, which has either a perceived or actual value to the organization

Note 1 to entry: An asset can be single entity, a collection of entities, an assembly of entities or a composition of entities.

EXAMPLE 1 Examples for physical entities are equipment, raw material, parts components and pieces, supplies, consumables, physical products and waste.

EXAMPLE 2 Software is an example of a digital asset.

EXAMPLE 3 A software license is an example of an intangible asset.

[SOURCE: IEC TR 63283-1:2022, 3.1.26, modified – the notes and example have been added]

3.1.2**Asset Administration Shell****AAS**

standardized digital representation of an asset

3.1.3**AAS interface**

interface of an AAS giving uniform access to information and services

EXAMPLE Examples for services are exploration services, asset services, and asset related services.

3.1.4**AAS responsible**

individual or organization having interest in an asset and governing an Asset Administration Shell

3.1.5**AAS user application**

software application which accesses an AAS via its AAS interface(s) for use by humans or for automatic processing

3.1.6**asset integration**

software or computing infrastructure, or both, needed to access asset services

3.1.7**asset service**

service that is provided by the considered asset

3.1.8**asset related service**

service that is not provided by the considered asset, but by the software or computing infrastructure, or both, outside of the considered asset

3.1.9**component**

product used as a constituent in an assembled product, system or plant

[SOURCE: IEC 61666:2010+AMD1:2021 CSV, 3.6)

3.1.10

concept

unit of knowledge created by a unique combination of characteristics

[SOURCE: IEC 61360-1:2017, 3.1.8]

3.1.11

concept repository

collection of entries that allows lookup by concept identifier and where relationships between entries can be described

[SOURCE: IEC 62832-1:2020, 3.1.5, modified – in the term "concept dictionary" has been replaced with "concept repository", in the definition "concept dictionary" has been deleted after "collection of", the second part of the definition, starting with "and where..." has been added and the note has been deleted]

3.1.12

concept repository entry

description of a concept containing, at a minimum, an unambiguous concept identifier, a preferred name, and a description

[SOURCE: IEC 62832-1:2020, 3.1.6, modified – in the term, "dictionary" has been replaced with "repository" and in the definition, "definition" has been replaced with "description"]

3.1.13

digital representation

information and services representing an entity from a given viewpoint

EXAMPLE 1 Examples of information are properties (e.g. maximum temperature), actual parameters (e.g. actual velocity), events (e.g. notification of status change), schematics (electrical) and visualization information (2D drawings, 3D drawing).

EXAMPLE 2 Examples of services are asset services (for example providing the history of the configuration data or providing the actual velocity) and asset related services (for example providing a simulation).

EXAMPLE 3 Examples of viewpoints are mechanical, electrical, or commercial characteristics.

3.1.14

end of life treatment

operation after a waste has been handed over to a facility for product and product part reuse, material recycling, energy recovery and residue disposal

Note 1 to entry: This includes dismantling, material separation and disposal.

SOURCE: IEC TR 62635:2012, 3.3, modified – insertion of Note 1 to entry]

3.1.15

entity

thing (physical or non-physical) having a distinct existence

[SOURCE: ISO/IEC 20924:2021, 3.1.18]

3.1.16**industrial sector**

grouping based on similar production processes, similar products, similar activities or similar behaviour in financial markets

EXAMPLE Examples are as follows. Health care technology, Environment–Health protection–Safety, Metrology and measurement–Physical phenomena, Testing, Mechanical systems and components for general use, Fluid systems and components for general use, Manufacturing engineering, Energy and heat transfer engineering, Electrical engineering, Electronics, Telecommunications–Audio and video engineering, Information technology, Image technology, Precision mechanics–Jewellery, Road vehicles engineering, Railway engineering, Shipbuilding and marine structures, Aircraft and space vehicle engineering, Materials handling equipment, Packaging and distribution of goods, Textile and leather technology, Clothing industry, Agriculture, Food technology, Chemical technology, Mining and minerals, Petroleum and related technologies, Metallurgy, Wood technology, Glass and ceramics industries, Rubber and plastic industries, Paper technology, Paint and colour industries, Construction materials and building, Civil engineering, Military affairs–Military engineering–Weapons, Domestic and commercial–equipment–Entertainment–Sports.

3.1.17**interface**

shared boundary between two entities defined by functional characteristics, signal characteristics, or other characteristics as appropriate

3.1.18**interoperability**

capability of two or more entities to exchange items in accordance with a set of rules and mechanisms implemented by an interface in each entity, in order to perform their respective tasks

EXAMPLE 1 Examples of entities include devices, equipment, machines, people, processes, applications, software units, systems and enterprises.

EXAMPLE 2 Examples of items include information, material, energy, control, assets and ideas.

3.1.19**instance asset**

specific asset that is uniquely identifiable

EXAMPLE Examples of instance assets are material, a product, a part, a device, a machine, software, a control system, or a production system.

3.1.20**property instance**

information consisting at least of the identifier of a property type and a property value

Note 1 to entry: The concept of type and instance applies to properties. If omitted, the term property refers to property types.

Note 2 to entry: The property instances have a value which can be provided by the manufacturer or another partner in the value chain. Sometimes, a property instance exists without a specific value, e.g. giving an existential statement.

Note 3 to entry: A property instance is called property-value pair in certain standards or data element in other standards.

3.1.21**property type**

defined parameter suitable for the description and differentiation of products

Note 1 to entry: The concept of type and instance applies to properties. If omitted, the term property refers to property types.

Note 2 to entry: The property types are defined in dictionaries (like IEC Common Data Dictionary or ECLASS), and they do not have a value. The property type is also called data element type in some standards.

[SOURCE: ISO/IEC Guide 77-2:2008, 2.18, modified – “type” was added to the term, the notes have been deleted and replaced with new notes]

3.1.22

role

set of characteristics that distinguish an entity's ability to exhibit a set of required behaviours

[SOURCE: ISO 18435-1:2009, 3.22]

3.1.23

service

distinct part of the functionality that is provided by an entity through interfaces

[SOURCE: IEC 60050-741:2020, 741-01-28]

3.1.24

software application

software functional element specific to the solution of a problem in industrial-process measurement and control

Note 1 to entry: A software application may be distributed among resources and may communicate with other software applications.

[SOURCE: IEC TR 62390:2005, 3.1.2; modified – in the term “application program” was replaced with “software application”]

3.1.25

Submodel

representation of an aspect of an asset

3.1.26

SubmodelElement

element of a Submodel

3.1.27

Submodel template

template for the representation of an aspect of an asset

3.1.28

Submodel template element

element of a Submodel template

3.1.29

system

interacting, interrelated, or interdependent elements forming a complex whole

[SOURCE: IEC TS 62443-1-1:2009, 3.2.123]

3.1.30

type asset

(abstract) representation of a set of instance assets with common characteristics and features

Note 1 to entry: The set of instance assets might exist or might not exist.

EXAMPLE Examples of type assets are type of material, a product type, a type of a part, a device type, a machine type, a type of software, a type of control system, a type of production system.

3.2 Abbreviated terms

AAS	Asset Administration Shell
AES	Advanced Encryption Standard
AML	AutomationML, Automation Markup Language
API	Application Programming Interface
BIM	Building Information Modelling
CAD	Computer Aided Design
CAEX	Computer Aided Engineering Exchange
CAM	Computer Aided Manufacturing
CDEL	Collection of Data Elements
CDV	Committee Draft for Vote
CE	Conformité Européenne
CNC	Computer Numerical Control
CRE	Concept repository entry
DC	Data Confidentiality
DF	Digital Factory
DI	Data Identifier
DIN	German Institute for Standardization
DIN SPEC	Specification published by German Institute for Standardization
ECCMA	Electronic Commerce Code Management Association
EDD	Electronic Device Description
EDDL	Electronic Device Description Language
EN	European Norm
ERP	Enterprise Resource Planning
ETIM	format to share and exchange product data based on taxonomic identification
EU	European Union
FB	Function Block
FDI	Field Device Integration
FDIS	Final Draft International Standard
FDT	Field Device Tool
FR	Foundational Requirement
GUID	Globally Unique Identifier
HTTP	Hypertext Transfer Protocol
IAC	Identification and Authentication Control
IACS	Industrial Automation and Control Systems
IC	Interface Class
ICT	Information and Communications Technology
ID	Identifier
IDTA	International Digital Twin Association
IE	Internal Element
IEC	International Electrotechnical Commission
IEC CDD	IEC Common Data Dictionary (IEC 61360 series)
IETF	Internet Engineering Task Force

IEV	International Electrotechnical Vocabulary
IH	Instance Hierarchy
IRDI	International registration data identifier
ISMS	Information Security Management System
ISO	International Organization for Standardization
IT	Information Technology
JEITA	Japan Electronics and Information Technology Industries Association
LOP	List of Properties
LV	Low Voltage
MDR	Metadata Registry
MES	Manufacturing Execution Systems
MOM	Manufacturing Operation Management
MOM	Manufacturing Operations Management
NID	Namespace Identifier
NSS	Namespace Specific String
OLOP	Operating List of Properties
OPC UA	OPC Unified Architecture maintained by the OPC Foundation
OT	Operational Technology
OTD	Open Technical Dictionary
PAS	Publicly Available Specification
PLM	Product Life Cycle Management
PS	Production System
RA	Resource Availability
RAI	Registration Authority Identifier
RC	Role Class
RDF	Restrict Data Flow
RFC	Request for Comments
RFID	Radio-Frequency Identification
SI	System Integrity
SL	Security Level
SOA	Service-Oriented Architecture
SSL	Secure Sockets Layer
SUC	System Unit Class
TCP	Transmission Control Protocol
TR	Technical Report
TRE	Timely Response to Event
TS	Technical Specification
UC	Use Control
UML	Unified Modeling Language
UN/ECE	United Nations Economic Commission for Europe
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name

UUID	Universally Unique Identifier
VI	Version Identifier
XML	Extensible Markup Language

3.3 Conventions

For some terms, such as Asset Administration Shell or SubmodelElement, start case or pascal case is used, in order to distinguish them from English text.

Subclauses in Clause 6 titled with "requirements" define requirements and recommendations.

4 Conceptual framework

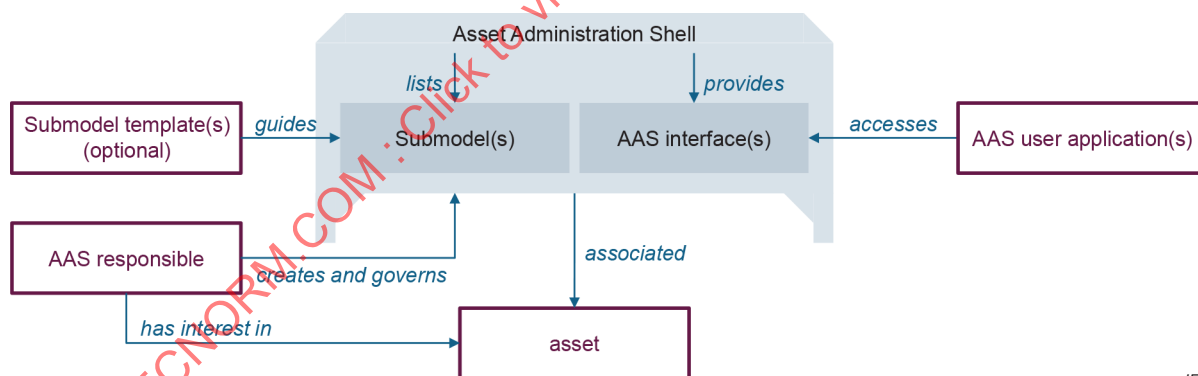
4.1 General

The conceptual framework describes the overall context of the Asset Administration Shell from an application perspective. This description is necessary for an overall understanding of the Asset Administration Shell (see also Annex B). Clause 4 focuses on permissions, possibilities and capabilities, although few requirements and recommendations are provided, in order to describe the overall application perspective.

4.2 Asset Administration Shell and related entities

4.2.1 General

Figure 2 depicts an overview of the Asset Administration Shell and related entities; it includes its main parts in grey and the related entities in purple. The grey colouring illustrates the scope of IEC 63278-1.



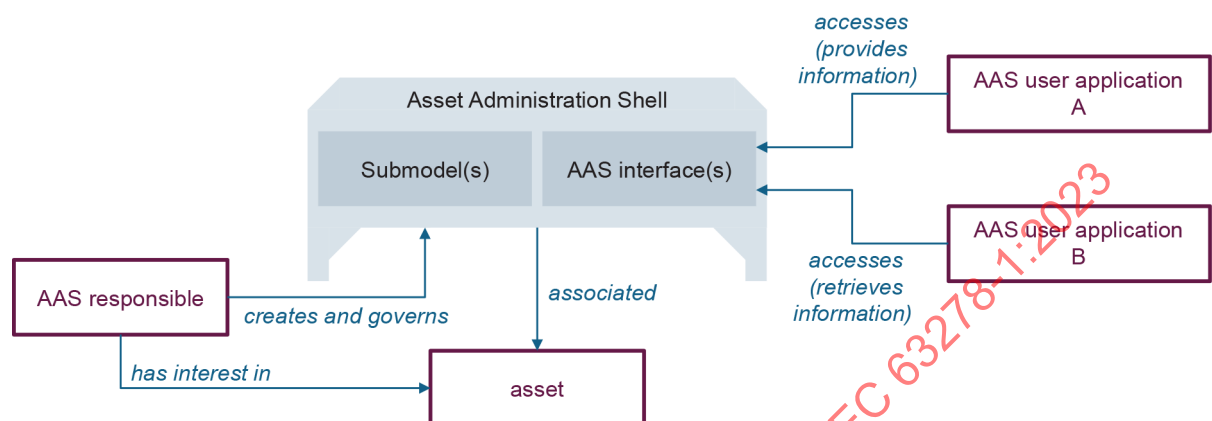
IEC

Figure 2 – Overview of Asset Administration Shell and related entities

The following relationships between the different entities shown in Figure 2 apply:

- an Asset Administration Shell is associated to an asset;
- an Asset Administration Shell provides AAS interface(s);
- there are one or more Submodels listed in an Asset Administration Shell;
- an AAS responsible has an interest in an asset, and based on this interest creates and governs an Asset Administration Shell;
- an AAS user application accesses an Asset Administration Shell via its AAS interface(s) for use by humans or for automated processing;
- a Submodel template may serve as guidance for a Submodel.

Figure 2 also illustrates that an AAS user application accesses information, which is represented by Submodels of the Asset Administration Shell, via the AAS interface of the Asset Administration Shell. Figure 3 extends this and shows how two different AAS user applications can exchange information about an asset. One AAS user application provides the information about the asset such as the delivery date via the AAS interface in the Asset Administration Shell and another AAS user application retrieves this information via the AAS interface. The AAS responsible may restrict the access individually for each AAS user application.



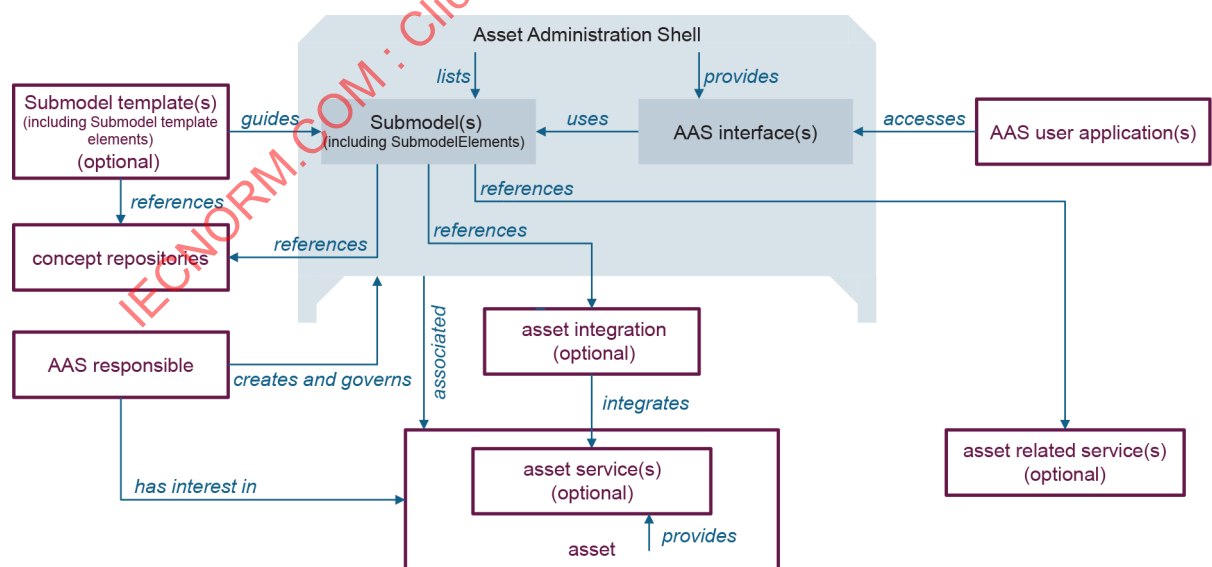
IEC

Figure 3 – Information exchange between AAS user applications

In addition, more complex scenarios are possible. For example, different AAS responsables may define their own different Asset Administration Shells for an asset (see Figure 6) and define the access by AAS user applications from their perspective. In this way, diverse use cases can be implemented for a cross-company exchange of information.

4.2.2 Detailed overview

Figure 4 shows a more detailed overview of the Asset Administration Shell.



IEC

NOTE The arrows in this figure only express that relationships do exist. Other parts of the IEC 63278 series can be more specific.

Figure 4 – Detailed overview of Asset Administration Shell and related entities

In addition to the relationships shown in Figure 2, Figure 4 shows the following additional relationships:

- Submodels reference concept repositories;
- a Submodel template may be used to serve as guidance for a Submodel and references concept repositories;
- asset can provide asset services;
- Submodels may reference asset services via an asset integration and may also reference asset related services.

All the elements shown in Figure 4 are explained in the subclauses 4.2.3 to 4.2.15.

4.2.3 Asset

An asset is valuable for an individual or an organization, that is why it is represented by an Asset Administration Shell. Only the AAS responsible determines what an asset is; this document does not make any specifications regarding what constitutes an asset.

An asset has a unique identifier.

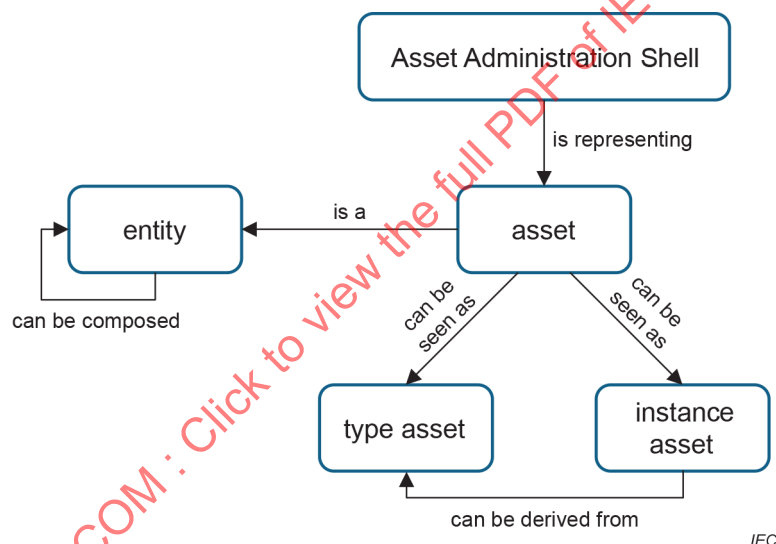


Figure 5 – Assets seen as type asset or instance asset

Figure 5 shows various kinds of assets and relationships between different assets. For example:

- an asset can be a simple entity, for example a single component, which is not composed of other entities;
- an asset can be composed of different entities. Such entities can be assets which are represented by an Asset Administration Shell or entities that are not represented by an Asset Administration Shell. For example, a machine can contain different drives which can be represented by Asset Administration Shells;
- there can be a derived-from relationship between a type asset and an instance asset, as an asset can be seen as a type asset or as an instance asset.

EXAMPLE For many assets, the distinction between type asset and instance asset can be applied. Manufacturers of products, automation equipment and production systems often use type assets to define a common set of characteristics offered to the market. A type asset can be used to derive specific instance assets with clearly defined characteristics. For example, a machine builder realizes a particular machine (instance asset) according to a type of machine (type asset).

Often, multiple instance assets are derived from the same type asset. Whether an asset is seen as type asset or instance asset is defined by the AAS responsible.

There are also technological relationships between assets, not shown in Figure 5, for example a pump feeds a tank, a drive is electrically connected to a power supply, or two parts are screwed together.

The Asset Administration Shell is therefore able to represent these various kinds of relationships (see 4.2.9).

NOTE The asset defined in this document is a super-set of the PS asset defined in the IEC 62832 series. In this document, asset also includes entities, which are not part of the production system.

4.2.4 AAS responsible

An AAS responsible defines the scope of the Asset Administration Shell and creates and governs the AAS. Typically, an AAS responsible defines only one Asset Administration Shell for an asset, focusing on assets as main reference points for structuring information.

An AAS responsible has a unique identifier.

The AAS responsible grants access rights to an AAS user application, i.e. an AAS user application might not be able to access all information or might not be allowed to execute all services offered via the AAS interface(s).

EXAMPLE Examples of AAS responsables are a machine manufacturer or a machine user (see Figure 6).

4.2.5 AAS user application

The purpose of the Asset Administration Shell is to enable two or more software applications to exchange information. These software applications are AAS user applications as shown in Figure 4.

An AAS user application has a unique identifier.

The development or execution of an AAS user application may be guided by a Submodel template.

4.2.6 Asset Administration Shell

The Asset Administration Shell gives uniform access to information and services of the asset or related to the asset, and thereby facilitates interoperability between AAS user applications.

An Asset Administration Shell provides AAS interface(s).

An Asset Administration Shell has a unique identifier.

There are Submodels listed in an Asset Administration Shell.

An Asset Administration Shell is created and governed by one AAS responsible and is related to one asset.

Relationships between different Asset Administration Shells may be defined:

- If there is a relationship between different assets and if these assets are represented by Asset Administration Shells, the relationship between the assets may be represented by a relationship between the respective Asset Administration Shells.
- Relationships between Asset Administration Shells may be defined even if there is no relationship between the associated assets.
- A derived-from relationship may be defined between the Asset Administration Shells of a type asset and of an instance asset.

These relationships are defined by SubmodelElements with the purpose of describing relationships (see 4.2.9).

An Asset Administration Shell shall be associated with an asset, which can be seen as a type asset or an instance asset.

There can be several AAS responsables having interest in the same asset, for example the manufacturer of a machine and the user of a machine (see Figure 6). This can result in several Asset Administration Shells with their respective AAS responsables associated with the same asset. However, this also means that different AAS responsables may assign different sets of characteristics to one asset.

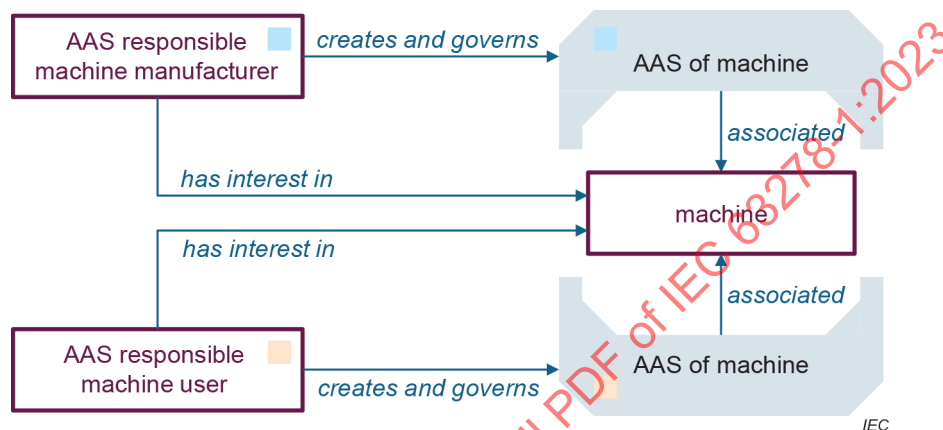


Figure 6 – Example of different Asset Administration Shells associated to the same asset

4.2.7 AAS interface

The purpose of the AAS interface is to enable an AAS user application to access information and services of the asset or related to the asset.

An Asset Administration Shell may have several AAS interfaces.

It is possible to allow several AAS user applications to access information of the same Asset Administration Shell via its AAS interface(s). The set of information that can be accessed can differ for different AAS user applications (see 4.2.4).

4.2.8 Submodel

A Submodel listed in an Asset Administration Shell represents information and describes services and thereby represents a specific aspect of an asset.

The Asset Administration Shell enables the execution of services described by a Submodel via the AAS interface and the AAS accesses asset related services and uses the asset integration to access asset services.

A Submodel has a unique identifier.

A Submodel may use a Submodel template as a template.

EXAMPLE An example of a Submodel is the digital nameplate. The European Union Directive 2006/42/EC [2] aims to standardize the market entry requirements for machines in the European Union and further related countries. Regarding nameplate the EU directive establishes the minimum requirements on information a nameplate must provide which state as follows:

- the business name and full address of the manufacturer and, where applicable, the authorised representative,
- designation of the machinery,
- the CE marking,
- designation of series or type,
- serial number, if any,
- the year of construction, that is, the year in which the manufacturing process is completed.

The digital nameplate combines all this information into a Submodel, see <https://www.zvei.org/presse-medien/publikationen/submodel-templates-of-the-asset-administration-shell-zvei-digital-nameplate-for-industrial-equipment-version-10>.

NOTE A Submodel as defined by this document is equivalent to a special type of Collection of Data Elements (CDEL) as defined in the IEC 62832 series. This document defines additional rules for a Submodel.

4.2.9 SubmodelElement

With a SubmodelElement different purposes can be pursued, for example:

- representing information about an asset;
- describing relationships between entities;
- describing services and enabling execution of services, for example asset services or asset-related services. How these services are executed is seen as an aspect of implementation and is out of the scope of this document;
- hierarchical structuring of Submodels.

NOTE Relationships are considered as a special kind of information.

A SubmodelElement may reference a Submodel template element as a template.

If a SubmodelElement does not reference a Submodel template element, a reference to an associated concept is necessary.

4.2.10 Submodel template

A Submodel template specifies the structure for a Submodel.

A Submodel template has a unique identifier.

Submodel templates guide the creation of Submodels. To be publicly accepted, these templates will often be defined according to existing international standards (see A.1.3).

4.2.11 Submodel template element

A Submodel template element specifies the structure for a SubmodelElement.

A Submodel template element specifies the structure depending on its purpose. The purpose is defined by the referenced concept, for example concepts describing:

- information about an asset;
- a relationship;
- a service;
- a structure.

A Submodel template element also specifies in which cardinality a SubmodelElement shall be present.

4.2.12 Concept repositories

Concept repositories serve to define common vocabularies and to describe relationships within these vocabularies. A common vocabulary is the basis to enable two or more software applications to exchange information and to mutually use the information that has been exchanged.

These concept repositories can be defined by standards, consortium specifications, manufacturer specifications and open community specifications (see Figure 7 and A.1.2). Concept repository entries can be referenced by Submodel templates and Submodels of an Asset Administration Shell. If there are changes in a concept repository, then it is the responsibility of the owner of the Submodel template and AAS responsible to respond appropriately. The AAS responsible, the owner of a Submodel template and the provider of a concept repository can be different organizations. Therefore, concept repositories, Submodel templates, and Submodels are managed independently.

EXAMPLE 1 IEC Common Data Dictionary (CDD) is an example of a concept repository owned and managed by the IEC. The digital nameplate is an example of a Submodel template managed by IDTA, whereas a manufacturing company uses the Submodel template as guidance for Submodels of their Asset Administration Shells.

NOTE Refer to IEC 62832-2:2020, Figure D.6 for similarities.

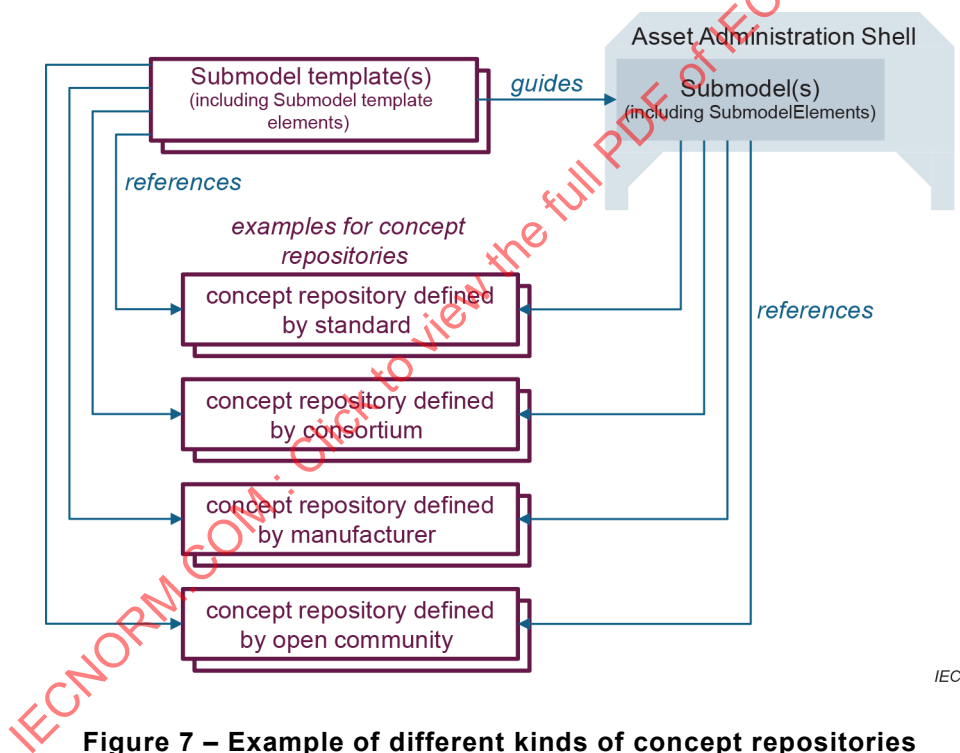


Figure 7 – Example of different kinds of concept repositories referenced by SubmodelElements

EXAMPLE 2 Examples of concept repositories are concept dictionaries like ECLASS or IEC CDD and ontologies like the open reference ontologies of the Industrial Ontologies Foundry, see <https://www.industrialontologies.org/>, or the Open Biological and Biomedical Ontology, see <https://obofoundry.org/>.

Concept repositories may be consulted by the AAS responsible during the design phase of a Submodel of an Asset Administration Shell and accessed by an AAS user application during the usage phase of an Asset Administration Shell.

- At the design phase, existing concept repositories may be input for AAS responsible to base the design of a Submodel of an Asset Administration Shell and to decide how to use and integrate a concept repository.

- In the usage phase, an AAS user application will access the information and services, which are represented by a Submodel. An AAS user application may also access concept definitions provided by the concept repository if they are referenced by SubmodelElements of a Submodel.

By using concept repositories, information and services provided by an Asset Administration Shell are based on standardized concept definitions.

EXAMPLE 3 A possible application of Asset Administration Shells is to provide information to AAS user applications to reason on finding matches between required capabilities and available skills. Products to be produced demand for capabilities, which are described resource-neutral. Industrial resources are represented by Asset Administration Shells, offering resource-neutral capabilities, available skills, and relationships between capabilities and skills. The "Capability for Industry Ontology" ontology describes these top-level concepts and relationships (see Figure 8). An AAS user application uses this information to reason, based on particular semantic knowledge of, for example, manufacturing processes, which are specified by a domain specific ontology like DIN 8580.

In Figure 9, an example of an Asset Administration Shell of a drilling machine is given. In one Submodel, a SubmodelElement for the capability deep hole is given, which refers to a CRE deep hole, for example specified by ECLASS. In another Submodel of the same Asset Administration Shell, a skill drilling is described and linked to the capability by a dedicated SubmodelElement with a purpose relationship. For precise control, some variable for the drilling depth is described, which refers to a concept repository entry also specified by ECLASS. Reasoning executed by an AAS user application is thought to set this drilling depth based on characteristics applied to the resource-neutral capability.

This example also illustrates how existing schemes like OPC UA specifications can be integrated to the Asset Administration Shell.

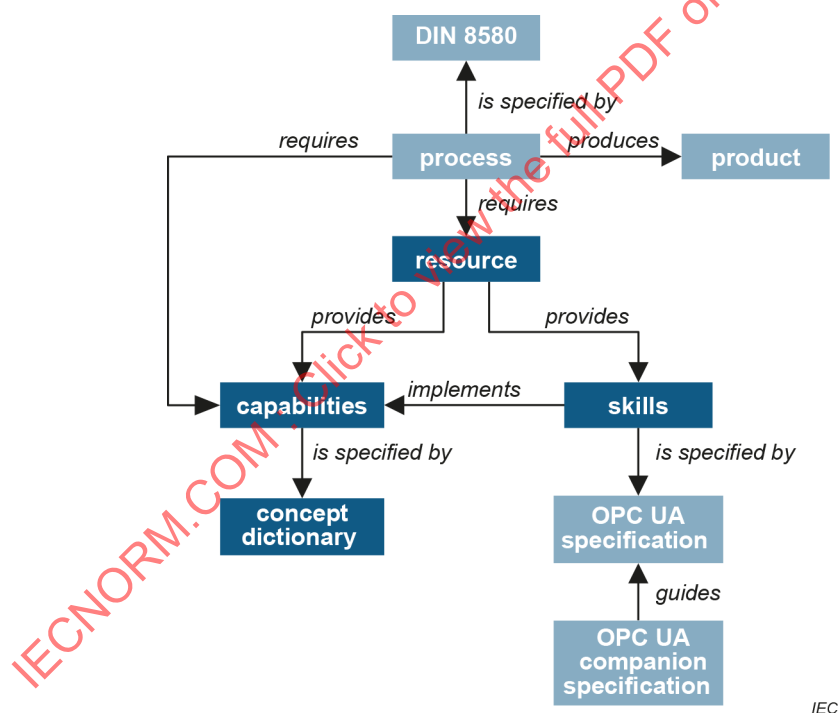


Figure 8 – Top level concepts and relationships of "Capability for Industry Ontology"

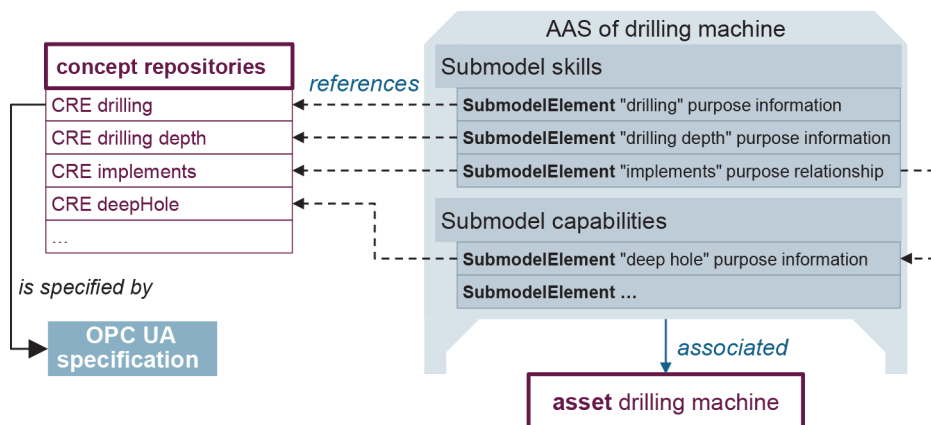


Figure 9 – Example of modelling by means of Asset Administration Shell

4.2.13 Asset integration

An asset integration serves as technical means to integrate and access, so that any asset service can be consistently accessed from an Asset Administration Shell.

EXAMPLE An additional edge device is integrated into a machine providing selected asset services of the machine. The edge device implements an asset integration to make these asset services available to the related Submodels (see Figure 10). Examples for asset services are providing the actual value of a machine and configuring the software of a machine. Examples for asset related services are providing the documentation for a machine or a simulation model of a machine, if the services are not provided by the asset itself, for example in the case of a purely mechanical component.

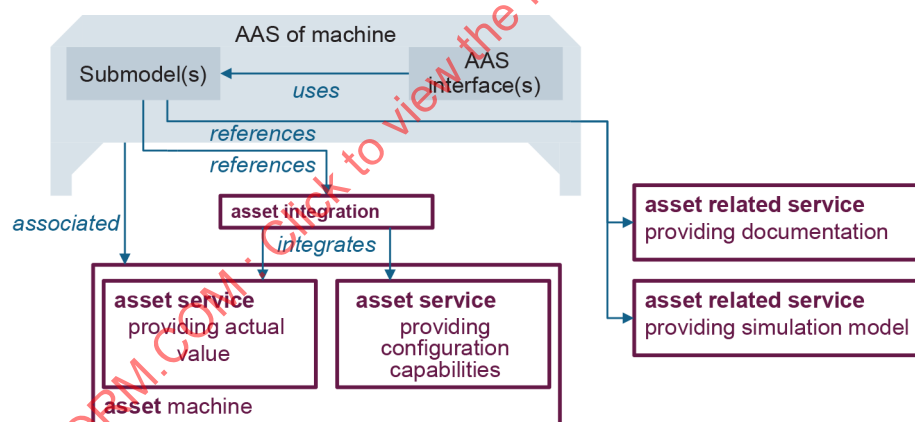


Figure 10 – Example of asset integration, asset services and asset related services

4.2.14 Asset service

An asset service serves to integrate into the Asset Administration Shell services that are provided by the asset itself.

4.2.15 Asset related services

An asset related service is a service that is logically assigned to the asset but is not provided by the asset itself.

4.3 Life cycle aspects of assets and Asset Administration Shells

This Subclause 4.3 describes characteristics of the application context of assets and AASs.

Assets and Asset Administration Shells have life cycles which should be considered individually. This is explained using the example of a machine (see Figure 11).

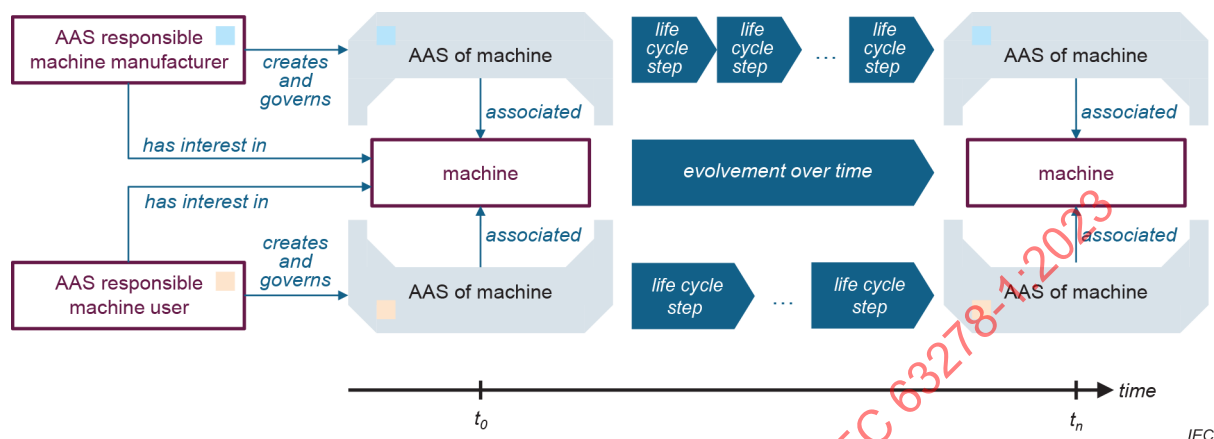


Figure 11 – Example of life cycle aspects of assets and Asset Administration Shells

A machine as physical asset changes over time, for example because of wear of a component, breakdown, reconfiguration, or minor damages. Some of these changes like wear are intrinsic to the asset and can even not be noticed by a machine user. Other changes, like breakdown, can be noticed by a machine user, others like reconfiguration are usually initiated by a machine user, and still others like minor damages can even not be relevant to the machine user. All these changes are summarized in Figure 11 as an evolution of the machine over time.

An Asset Administration Shell changes over time, for example the AAS responsible initiates the addition of a Submodel or Asset related service, an AAS user application modifies the information represented by some SubmodelElement, and an asset service provides updated information about the asset resulting in an update of information represented by some SubmodelElements. These changes are not intrinsic but result from explicit effects of entities related to the Asset Administration Shell according to Figure 4. These changes are shown in Figure 11 as discrete life cycle steps. As different Asset Administration Shells are implemented by different AAS responsables (light blue and light orange rectangles), these life cycle steps are specific for each Asset Administration Shell.

The Asset Administration Shell of the machine user and the Asset Administration Shell of the machine manufacturer can be created at different times and then typically coexist. These Asset Administration Shells are independent and therefore it is not guaranteed that these Asset Administration Shells are logically consistent. Synchronization between these Asset Administration Shells depends on the underlying use cases and is under the responsibility of the corresponding AAS user applications (see also 4.4).

The following example illustrates the interests of different AAS responsables and their views on different assets and associated Asset Administration Shells. It exemplifies a real-life value chain between a product manufacturer and a product user who integrates the product, for example an automation component, as a component into the production system (see Figure 12).

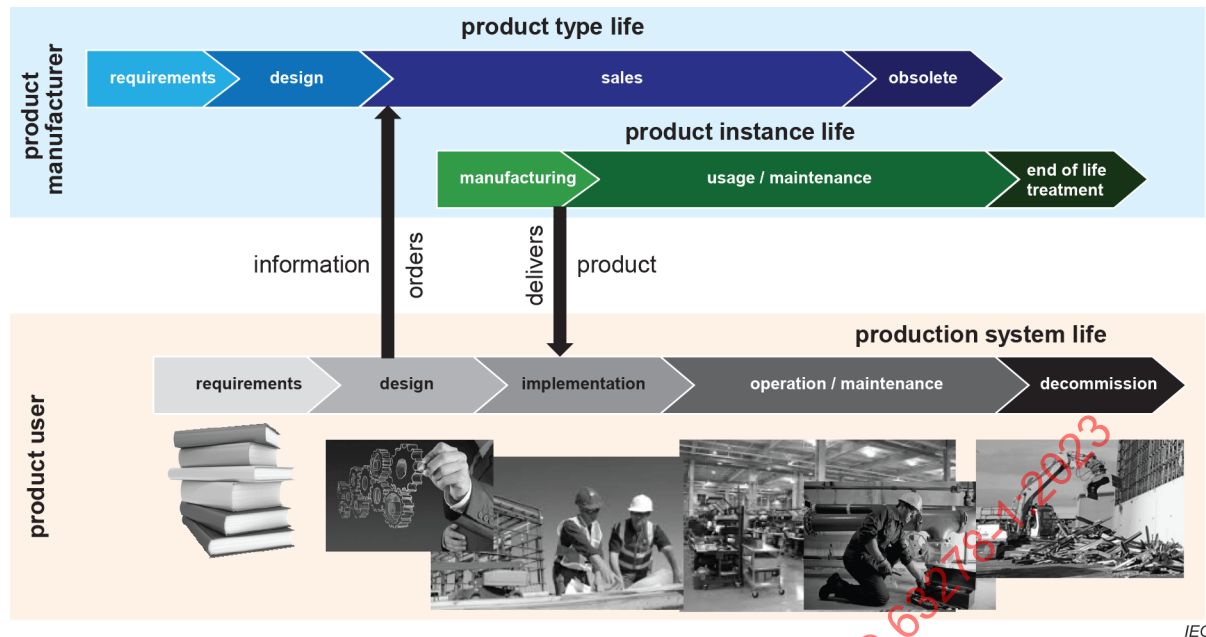


Figure 12 – Example of assets in product and production system life cycles

NOTE IEC 62890 defines product types and instances, the asynchrony of their life cycles, the implications to the production system life cycle and the duality of product and component.

The product manufacturer (as AAS responsible) considers and manages different assets over their life cycles:

- The product type that has a history with different versions.
- In parallel, the product manufacturer produces a number of product instances of the product type and its versions.

As shown in the example in Figure 12, the product instance is represented by an Asset Administration Shell of an instance asset, and the product type is represented by an Asset Administration Shell of a type asset.

The product user will incorporate such product as a component in the production system. Also, the product user (as AAS responsible) will deal with the following assets over their life cycles:

- Role asset: The product user will define the role that will be assumed by the product. This role will be included in the overall digital descriptions of the production system and managed along the life cycle of the production system. This role will also be used for deciding the ordering of the right product instance from the right product manufacturer.
- Product instance asset: In parallel, the product user will manage the product instance (component) delivered by the product manufacturer over its life cycle.

If roles are considered as assets, they can be represented by Asset Administration Shells. The product user will represent the product instance (component) by an Asset Administration Shell for an instance asset which will comprise information provided by the product manufacturer and role information.

EXAMPLE During the requirements phase it was found that a specific function for transporting parts to the machine was needed. The required function with its parameters (e.g., count, weight, and size of the parts) can be described as a role. The description of the role constitutes a value in itself, because of the required effort to develop the description and because several experts from different domains can be involved in providing the description. To facilitate exchange of information and for maintaining this information for later phases in the system life cycle (e.g., for maintenance), the responsible requirements manager can decide to represent the role by an Asset Administration Shell. The role description is later the base for selecting the optimum design for transport (e.g., a conveyor belt or a set of automated guided vehicles).

Figure 13 illustrates the Asset Administration Shells for the assets illustrated in Figure 12. The asset "product instance" is derived from the asset "product type" (see Figure 5) and Figure 13 also illustrates that this derived-from relationship between assets is represented by a derived-from relationship between Asset Administration Shells, see 4.2.6.

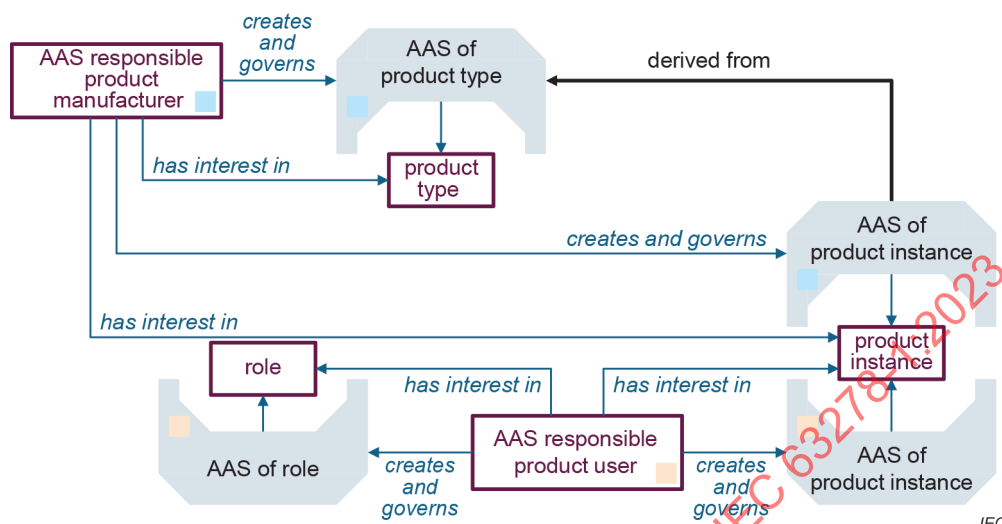


Figure 13 – Example of Asset Administration Shell in product life cycle

4.4 Example for an overall Asset Administration Shell scenario

Subclause 4.4 gives an example and integrates the various perspectives described in 4.3 into an overall scenario. Figure 14 illustrates how the life cycle perspective illustrated in Figure 11, the value exchange perspective illustrated in Figure 12 and the assets and Asset Administration Shells perspective illustrated in Figure 13 can be combined in an overall scenario.

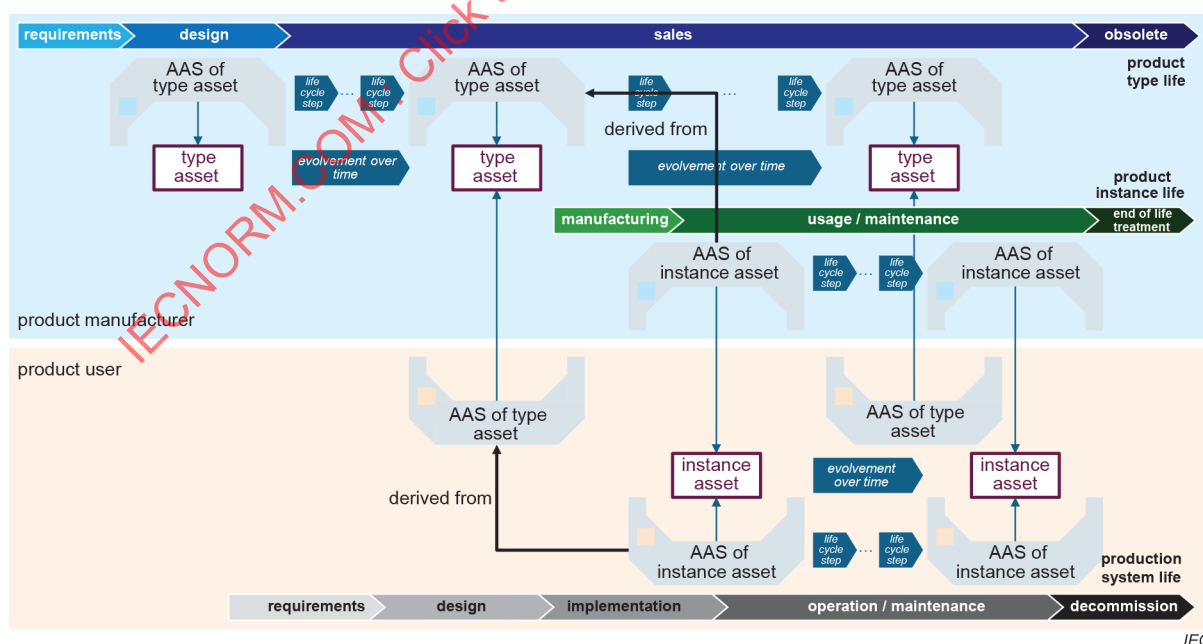


Figure 14 – Illustration of an example of an overall Asset Administration Shell scenario

Figure 14 is based on the following design principles:

- The timeline runs from left to right.
- For reasons of clarity, the AAS responsible has been omitted, as it is sufficiently expressed by the colouring in light blue and light orange.
- Figure 14 shows two assets, one asset is called "type asset" and the other is called "instance asset". Two Asset Administration Shells are associated to each asset, one Asset Administration Shell created and governed by the product manufacturer and the other Asset Administration Shell created and governed by the product user.
- At a certain point in time, the corresponding status of the two assets in their life cycle and the current representation of the asset by the Asset Administration Shells is shown. Figure 14 does not show that some infrastructure for the Asset Administration Shell can provide capabilities to retrieve the representation of an asset at a point in time in the past.
- A product manufacturer can maintain for a product type different versions over the time. This additional complexity is not shown in Figure 14. This can be modelled by considering each version as a separate asset and modelling the relationships between these different assets. It is assumed that the derived-from relationship between the corresponding Asset Administration Shells remains in place over the life cycle of the Asset Administration Shells.

Figure 14 shows a product which is integrated in a production system of the product user. The purpose of the various Asset Administration Shells is as follows:

- Asset Administration Shell of type asset created and governed by the product manufacturer: The product manufacturer typically has an interest to manage a product type over the life cycle.
- Asset Administration Shell of type asset created and governed by the product user: In this scenario it is assumed that the product user has an interest to manage the product type over time, because the product user wants to manage for example their own delivery conditions agreed on with the product manufacturer, their own demands regarding the ordering of several product instances of the product type or some specific technical content regarding the product type.
- Asset Administration Shell of instance asset created and governed by the product manufacturer: The product manufacturer typically has an interest to manage each product instance individually over the life cycle, especially if the product instance is individually configured based on the needs of a product user. Examples are the management of the specific configurations sold to the various product user or the offering of spare parts, which are requested for a specific product instance of a product user.
- Asset Administration Shell of instance asset created and governed by the product user: The product user typically has an interest to manage a product instance over the life cycle, for example for maintenance management of the product instance or collection of usage information of the product instance to optimize the usage of the product instance. Sometimes a product user does not want to disclose the usage information of a product instance to the product manufacturer.

Figure 15 extends Figure 14 by showing the value exchange between the product manufacturer and product user:

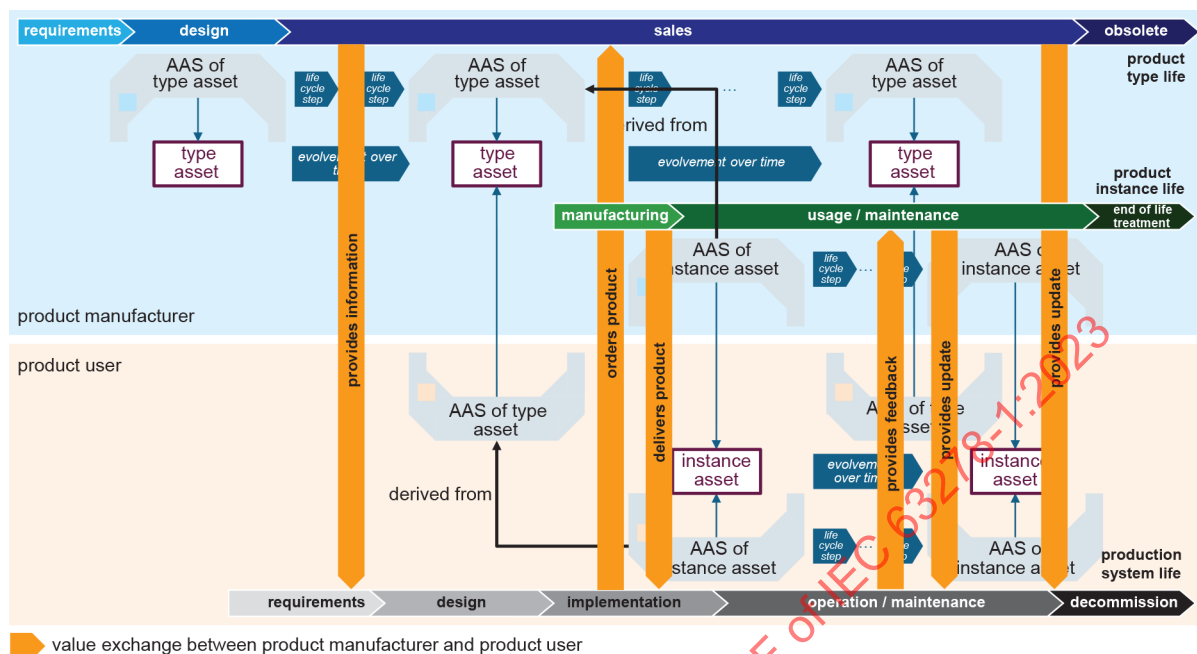


Figure 15 – Illustration of value exchange in overall scenario

- Product manufacturer provides information about product type: Based on this generally available information represented in the Asset Administration Shell of the product manufacturer, a product user creates its own Asset Administration Shell of the product type to manage information of the product type for their own purpose.
- Product user orders product from product manufacturer: Based on the information in the Asset Administration Shell of the product type of the product user, the product user orders a specific product. Typically, at this time at the latest, the product user creates an Asset Administration Shell of a product instance in which the product user for example manages information about the order. Typically, the product manufacturer also creates its own Asset Administration Shell of the product instance at this time, in which the product manufacturer for example manages information about the manufacturing process of this specific product instance.
- Product manufacturer delivers product to product user: In addition to the physical product, the product manufacturer provides the product user with further information about the product instance supplied. The product user takes over the supplied information of the Asset Administration Shell of the product manufacturer and expands the product user's Asset Administration Shell of the product instance with this information. This illustrates that the product manufacturer does not disclose all information to the product user.
- Product user provides feedback to product manufacturer: Examples of such feedback are the ordering of a spare part or the provision of usage information about the product to take advantage of the data-driven services offered by the product manufacturer in return. The product manufacturer takes over the supplied information of the Asset Administration Shell of the product user and expands the product manufacturer's Asset Administration Shell of the product instance with this information. This illustrates that the product user does not disclose all information to the product manufacturer.
- Product manufacturer provides updated information about product instance to product user: Examples for such updated information are the provision of a product instance-specific updated document, a product instance-specific software update or a data-driven service. The product user integrates this updated information into their own Asset Administration Shell of the product instance. This illustrates that the synchronization of information of different Asset Administration Shells of the same asset depends on the specific interests of the different AAS responsible for the Asset Administration Shells.

- Product manufacturer provides updated information about product type: Examples for such updated information are the provision of updated documentation or software updates or changes to the delivery conditions. The product user integrates this updated information into their own Asset Administration Shell of the product type and possibly also into their own Asset Administration Shell of the product instance.

5 Identifiers

5.1 Needs

Asset Administration Shell and its elements can be accessed by AAS user applications. Such AAS user applications are created and used by different organizations worldwide (see 4.2.1, 4.2.3).

SubmodelElements with a purpose relationship can be used to express relationships between these Asset Administration Shells and their elements (see 4.2.9). The following elements require unique identifiers: Asset Administration Shells, Submodels, assets, used concepts, AAS responsible, AAS user application, Submodel templates (see 4.2). Taken together with the global namespace, these identifiers shall be globally distinct.

NOTE Other elements, such as SubmodelElements (see 4.2.9), are hierarchically structured by other elements, such as Submodels, and therefore do not require a globally distinct identifier.

EXAMPLE A machine is qualified for the food industry. It contains multiple drives from different component manufacturers. The machine builder and the component manufacturers are part of the global market. In order to pursue the qualification, the Submodels for technical data of each of these drives is accessed and assessed to check whether the lubrication is conformant with the regulations in the food industry.

5.2 Determination of identifiers

5.2.1 General

For the following international and freely available standards and specifications, appropriate variants of identifiers are determined (see Figure 16).

There are two categories of identifiers for which the requirements are different:

- identifiers, which were provided by external roles, and are simply to be used, such as globally distinct identifiers for entries in concept repositories;
- identifiers, which are to be created by the AAS responsible (or respective AAS user application), such as globally distinct identifiers for assets, Asset Administration Shells and Submodels.

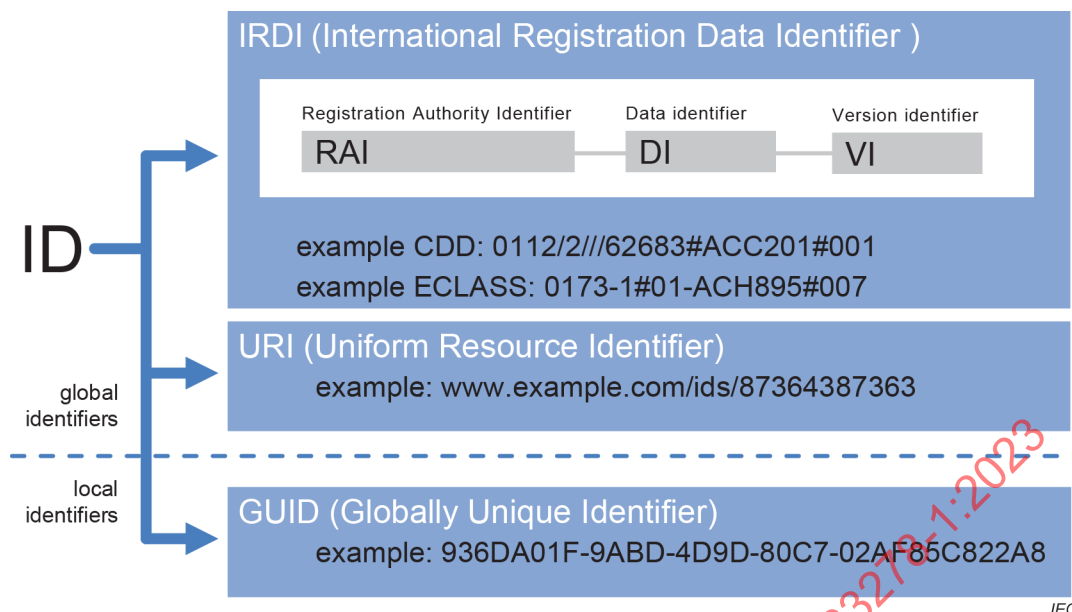


Figure 16 – Different identifiers for globally distinct identifiers and local identifiers

5.2.2 Globally distinct identifiers for concepts by IRDIs

ISO/IEC 11179-6, ISO/IEC 6523 and ISO 29002 establish an identification scheme for globally distinct identifiers, named IRDI. The uniqueness of an IRDI is determined by the combination of the values of three components:

- Registration Authority Identifier – RAI;
- Data Identifier – DI;
- Version Identifier – VI.

5.2.3 Globally distinct identifiers by URIs

By means of the URI schema (IETF RFC 3986 [3]), domain and path, type-secure, globally distinct and distributed volumes of identifiers can be realized.

URIs are suitable as identifiers for concept repository entries, assets, Asset Administration Shells, Submodels, AAS responsables and AAS user applications.

EXAMPLE 1 An organization is creating identifiers using uniform resource locator (URL). Identifiers will have the hostname which is registered to that organization. An enumeration procedure for the paths following the hostname within the organization will ensure that only unique paths are enumerated for particular identifiers. Taking hostname and paths together will ensure globally distinct identifiers by URL.

EXAMPLE 2 An organization is creating identifiers using uniform resource name (URN). A distinct namespace identifier (NID) is chosen for the URNs, e.g. based on the hostname of the organization. For the namespace specific string (NSS), an enumeration procedure is established as above. Taking NID and NSS together will ensure globally distinct identifiers by URN.

5.2.4 Local identifiers

Local identifiers are suitable if they are not accessed by AAS user applications used by organizations different from the AAS responsible. For example, these can be globally unique identifiers (GUIDs) or can be identifiers of manufacturers' internal data elements.

However, for identifiers of manufacturers' internal data elements, the use of IRDIs is recommended as they are unique when a company-specific International Code Designator (ICD) is assigned.

6 Asset Administration Shell structure

6.1 General

In Clause 6 a general structure for an Asset Administration Shell is defined by stating requirements, recommendations, and permissions. The focus is on the grey parts in Figure 4. If an implementation of the Asset Administration Shell satisfies these requirements, recommendations, and permissions, usage of the Asset Administration Shell as described in Clause 4 is possible and interoperability is achieved.

NOTE For a discussion of interoperability see 0.3.

The general structure is based on established conceptual works in both automation and ICT technologies. The defined structure is prepared for future developments regarding the relevant aspects such as horizontal integration and vertical integration. The intention of Figure 17 is to demonstrate the structure with an example.

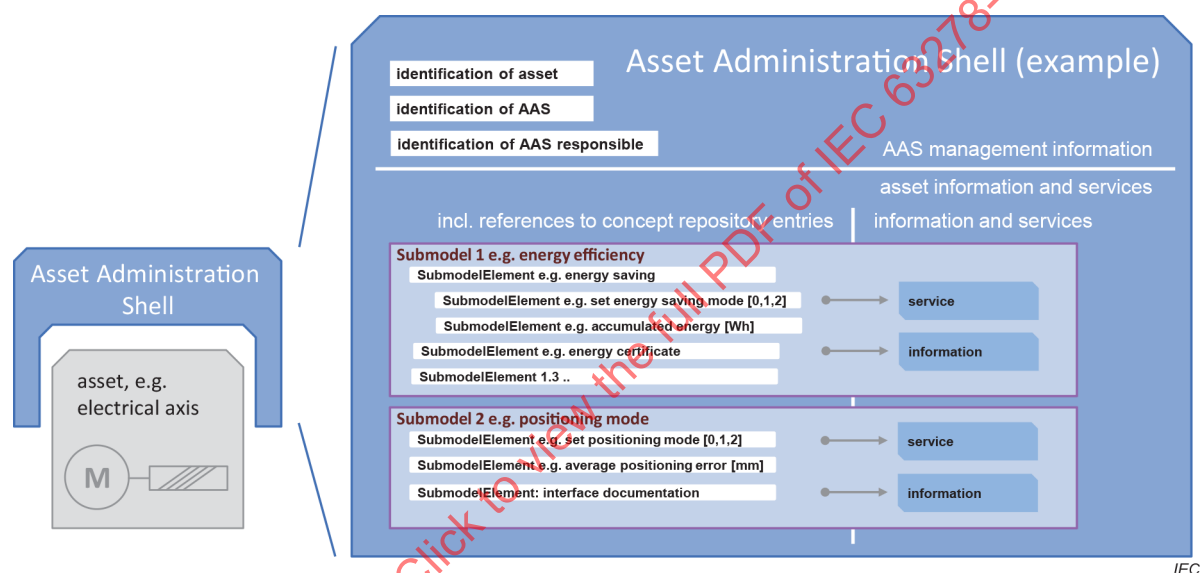


Figure 17 – Example of an Asset Administration Shell demonstrating the general structure

AAS management information in Figure 17 is used to identify Asset Administration Shells, the related asset and the related AAS responsible and to manage the Asset Administration Shell over its life cycle (see 4.3).

NOTE Depending on the kind of asset, which is represented by the Asset Administration Shell, the AAS management information is similar to the DF asset class header or to the DF asset header in the IEC 62832 series. In IEC 62832 a header provides at least an element identifier, version number, preferred name and description. The AAS management information provides additional information specific to the Asset Administration Shell.

The asset information and services in Figure 17 include individual Submodels. As shown by Figure 17, each Submodel has hierarchically organised SubmodelElements each of which refer to individual information and services (depicted by the blue rectangles in Figure 17). Services can be asset services (via asset integration) or asset related services.

The AAS management information and the asset information and services are safeguarded by provisions for information security adequate for the industrial application of the assets involved (see 6.3).

6.2 Requirements associated to Asset Administration Shell

6.2.1 General

The provisions in subclause 6.2 are derived from the analysis of the conceptual framework in Clause 4. Provisions are separated in subclauses followed by further statements, notes, and examples.

6.2.2 Asset Administration Shell

6.2.2.1 Requirements related to information management

6.2.2.1.1 Requirements

The Asset Administration Shell shall have a globally distinct identifier. For the identifier, URI shall be used.

An Asset Administration Shell shall have a globally distinct identifier of the asset it is representing. For the identifier, URI shall be used.

An Asset Administration Shell should have a distinct identifier for the AAS responsible.

An Asset Administration Shell shall provide a defined set of AAS management information.

An Asset Administration Shell shall support the management of AAS management information over the life cycle.

6.2.2.1.2 Rationale and guidance

The identifier of the Asset Administration Shell is different from the identifier of the asset. The Asset Administration Shell shall be identified uniquely, because several AAS responsables can create different Asset Administration Shells for the same asset (see Figure 6).

NOTE 1 Depending on the kind of represented asset, the identifier of the Asset Administration Shell is equivalent to the element identifier of a DF asset class (asset kind = type) or to the element identifier of a DF asset (asset kind = instance) according to the IEC 62832 series.

NOTE 2 It is possible to use a GUID and express this as an URI.

The association between assets and Asset Administration Shells is very important. These associations shall not break, even if they are saved in digital repositories or saved in a manner which spans all value-added partners.

NOTE 3 The unique identifier of the asset can be made available by the asset itself, e.g. on its nameplate.

NOTE 4 The identifier of the asset is equivalent to the PS asset identifier of the DF asset in accordance with the IEC 62832 series.

NOTE 5 The use of URI as identifiers of assets is described in IEC 61406-1:2022 and in IEC 61406-2 CDV.

The identifier for the AAS responsible is intended to recognize who is responsible for a specific Asset Administration Shell.

The provisions for information management are necessary to manage an Asset Administration Shell over its life cycle.

NOTE 6 AAS management information is, for example, the asset's globally distinct identifier, version information, timestamps of data or of structural changes, etc.

6.2.2.2 Requirements related to structure

6.2.2.2.1 Requirements

An Asset Administration Shell shall have Submodels including SubmodelElements and one or more AAS interfaces.

6.2.2.2.2 Rationale and guidance

To support interoperability among several AAS user applications, a well-known structure of information and AAS interfaces are important.

6.2.2.3 Requirements related to asset classification

6.2.2.3.1 Requirements

An Asset Administration Shell shall describe if an asset is seen as type asset or instance asset or whether this kind of classification is not applicable to the asset.

An instance asset shall be represented by an Asset Administration Shell for an instance asset, and a type asset shall be represented by an Asset Administration Shell for a type asset.

6.2.2.3.2 Rationale and guidance

Figure 5 and the explanation in 4.2.3 provide rationale and guidance with respect to asset classification.

6.2.3 Submodel

6.2.3.1 Requirements general

6.2.3.1.1 Requirements

An Asset Administration Shell shall support Submodels depending on the needs of the AAS responsible.

An Asset Administration Shell shall support the management of a Submodel over its life cycle.

6.2.3.1.2 Rationale and guidance

Submodels are used to structure the information and services of an Asset Administration Shell into distinguishable parts. Each Submodel refers to a well-defined domain or subject matter.

As assets evolve over time, the digital representations shall be maintained over the lifetime as well (see 4.3 and 4.4).

6.2.3.2 Requirements related to structure

6.2.3.2.1 Requirements

A Submodel shall support the organization of SubmodelElements into hierarchical structures.

6.2.3.2.2 Rationale and guidance

Submodels organize SubmodelElements. In order to cope with the volume of SubmodelElements, a means of hierarchical structuring is required.

NOTE 1 A Submodel, which features SubmodelElements such as properties, corresponds to a CDEL according to the IEC 62832 series and can have a hierarchical sub-structure. A CDEL definition can describe the semantics of a Submodel.

NOTE 2 A collection of SubmodelElements corresponds to a CDEL of the IEC 62832 series for hierarchical structuring.

6.2.3.3 Requirements related to semantics

6.2.3.3.1 Requirements

The concept of the Asset Administration Shell should support the creation of Submodels guided by predefined Submodels templates, in addition to Submodels, where its SubmodelElements reference to concept repository entries of concept repositories.

If guided by a Submodel template, the Submodel shall reference the Submodel template by its globally distinct identifier.

6.2.3.3.2 Rationale and guidance

Submodel templates serve to guide a Submodel causing this Submodel to feature a guaranteed set of SubmodelElements. Submodel templates will therefore have their own life cycle, even independent of the life cycle of the references concept repositories, because the owner of the Submodel template can be different from the AAS responsible and the provider of concept repositories.

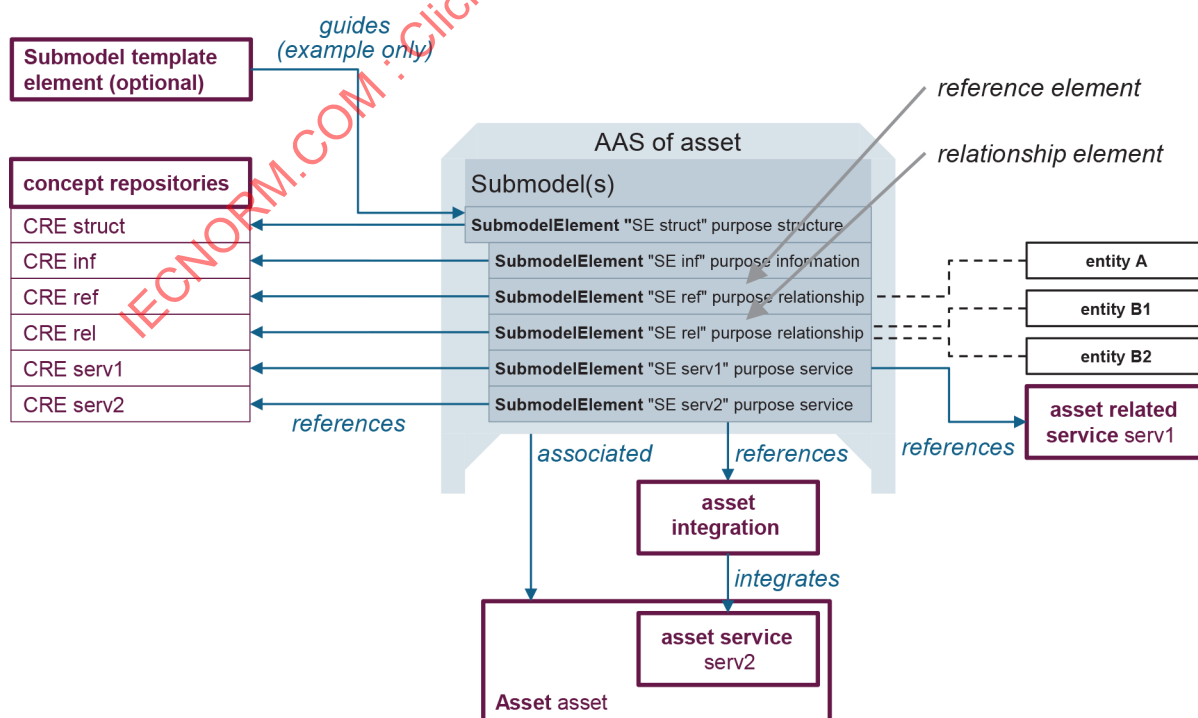
NOTE 1 Such Submodels templates can standardize information and services, e.g. for a digital nameplate of assets, data sheets of components or handover documentation of equipment.

NOTE 2 Additional documents can define how Submodel templates will be standardized in international standardization.

6.2.4 SubmodelElements

6.2.4.1 General

Based on Figure 4 the different aspects of SubmodelElements are illustrated in Figure 18 to serve as guidance for the various requirements for SubmodelElements.



IEC

Figure 18 – Illustration of different aspects of SubmodelElements

6.2.4.2 Requirements general

6.2.4.2.1 Requirements

A SubmodelElement shall support the description of information.

A SubmodelElement shall support the description of the reference to a clearly identified entity.

A SubmodelElement shall support the description of a relationship between two clearly identified entities.

A SubmodelElement shall support the description of the reference to an asset service via an asset integration.

A SubmodelElement shall support the description of the reference to an asset related service.

A SubmodelElement shall support the organization of SubmodelElements into hierarchical structures.

6.2.4.2.2 Rationale and guidance

According to Figure 17, the structure of Submodels and SubmodelElements within an Asset Administration Shell serves as a clearly defined table of contents for all information and services. SubmodelElements are of uniform structure and are thus providing a very stable source of information.

In the manufacturing domain information and services are of a large variance and are often of very complex structure. Therefore, SubmodelElements refer to these complex data, digital models, and services in order to expose these to the AAS interface(s) in a uniform approach.

NOTE 1 This approach relies on an extended understanding of the IEC 61360 series property concept.

Information and services often have relationships to other entities. To be able to safely navigate through these relationships, it is necessary to clearly identify the referenced entities. This includes, without claiming to be complete, for example: the identifier of the entity (see Clause 5), information about the type of entity (e.g., an asset, an Asset Administration Shell, a Submodel or a SubmodelElement) or the identifier of an Asset Administration Shell, if entities of another Asset Administration Shell are to be referenced. This information can be described by appropriate SubmodelElements. Details will be defined in IEC 63278-2.

NOTE 2 For referencing SubmodelElements, a unique identifier of a SubmodelElement can be achieved by using a unique identifier of a Submodel chained together with identifiers of subordinate SubmodelElements and an identifier of the SubmodelElement itself.

A relationship between two SubmodelElements combines two references with a semantic identifier for a concept repository entry.

6.2.4.2.3 Examples of the requirements

SubmodelElements require the representation of characteristics of assets for multiple purposes. Table 1 shows examples for the purposes supported by SubmodelElements.

Table 1 – Examples for categorization of SubmodelElements

Specialization of SubmodelElement	Description	Purpose
Collection of SubmodelElements	Represents hierarchical structures, sets and lists of SubmodelElements.	Structure
Property instance	Represents characteristics referring to property types, e.g. in standardized dictionaries.	Information
File	Represents complex information of specific data format relating to a single concept repository entry.	Information
Reference element	Represents a comprehensible reference to another entity, such as an asset, Asset Administration Shell, Submodel or SubmodelElement.	Relationship
Relationship element	Represents a triple knowledge: subject, predicate, and object. Subject and object are described by references. The predicate is described by the element itself and its referred concept repository entry.	Relationship
Service element	Represents a service and enables the execution of a service, e.g. asset services or asset-related services.	Service

NOTE 1 The representation of instances of properties in accordance with the IEC 61360 series (see Clause A.2) and the ISO 22745 series (OTD) are based on the format for DataElement as defined by IEC 62832-2. For different types of information and functions various complementary data formats and methods of access are used.

One of the typical relationships used to describe a composite asset is the "composed of" relationship, see Figure 5. Figure 19 illustrates an example for a "composed of" relationship of an assembly of electrical axes.

EXAMPLE A set of mechanical axes and grippers constitutes an encapsulated assembly, which is represented by a dedicated Asset Administration Shell and references its subordinate Asset Administration Shells for the individual mechanical axes and grippers. Each Asset Administration Shell exposes an individual AAS interface to be used by AAS user applications.

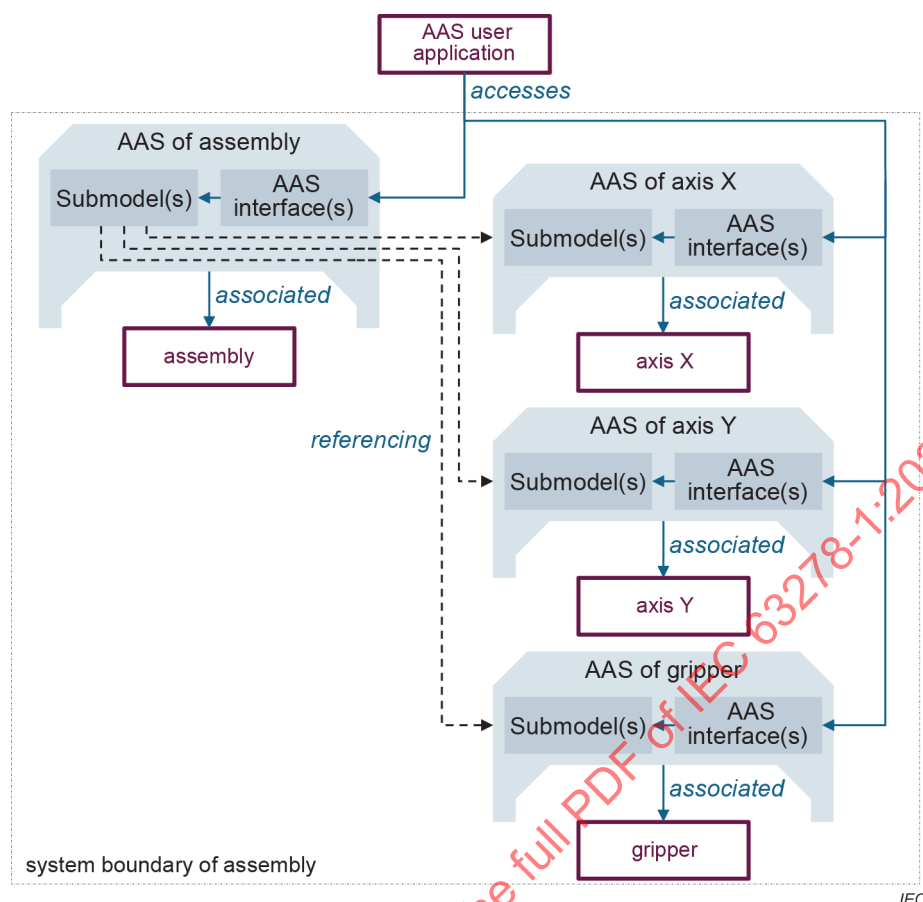


Figure 19 – Asset Administration Shells of a representative assembly of electrical axes

Another typical relationship used to describe a relationship between an Asset Administration Shell representing an instance asset and the Asset Administration Shell representing the corresponding type asset is the "derived-from" relationship. The "can be derived from" relationship between the instance and type asset shown in Figure 5 and illustrated in Figure 13 as a relationship between the corresponding Asset Administration Shells is represented by a SubmodelElement with a purpose relationship referring to a concept repository entry "derived-from".

Typically, selected Submodels of the Asset Administration Shell representing the type asset are taken over by the Asset Administration Shell representing the instance asset. However, an Asset Administration Shell representing an instance asset can have additional Submodels and can also change the values of the SubmodelElements taken over.

NOTE 2 For example this can also mean that the representations of a type asset are extended over the life cycle or, for the Asset Administration Shell of an instance asset, Submodels or SubmodelElements are added, amended or deleted depending on (maintenance) activities of the respective asset.

6.2.4.3 Requirements related to semantics

6.2.4.3.1 Requirements

A SubmodelElement shall support the specification of a reference to a concept repository entry provided by a concept repository.

For identifiers of concept repository entries, any identifier type that provides a globally distinct identifier shall be supported.

An Asset Administration Shell shall support different concept repositories to be referenced by different SubmodelElements.

The concept of the Asset Administration Shell should support the creation of a SubmodelElement guided by a predefined Submodel template element.

6.2.4.3.2 Rationale and guidance

References to well-known concept repositories are important for semantic interoperability.

NOTE 1 Well-known data dictionaries such as IEC CDD, OTD, or ECLASS use IRDI.

NOTE 2 For ontologies found on the internet, identifiers according to URI are used.

The need for different concept repositories is explained in 4.2.12.

The guidance of Submodel templates elements follows the same principles as the guidance of Submodel templates (see 4.2.10).

6.2.5 AAS interface

6.2.5.1 Requirements

All Asset Administration Shells shall provide a set of standardised AAS interfaces to enable AAS user applications to access the information and to access and execute the services they are offering in a uniform way.

All Asset Administration Shells shall provide a set of standardised AAS interfaces to enable AAS user applications to create, delete and update an Asset Administration Shell and the Submodels and SubmodelElements in a uniform way.

All Asset Administration Shells shall provide a set of standardised AAS interfaces to enable AAS user applications to retrieve an Asset Administration Shell not only by the unique identifier of the asset it is representing but also by alternative identifiers of the same asset. These alternative identifiers shall be unique in a given namespace only, but not globally.

6.2.5.2 Rationale and guidance

An AAS interface makes the information and services of the Asset Administration Shell available externally. Some examples are (the list is not exhaustive):

- read/write SubmodelElements with purpose information;
- execute SubmodelElements with purpose service;
- explore Submodels and SubmodelElements with purpose structure and relationship;
- create, delete and update for an Asset Administration Shell, a Submodel and a SubmodelElement;
- retrieve an Asset Administration Shell.

NOTE 1 The standardized AAS interface will be defined in a future part of the IEC 63278 series.

NOTE 2 Based on these basic concepts provided by the AAS interfaces an AAS user application can implement more complex concepts like copying, mirroring, and versioning. Some of these complex concepts are illustrated in 4.4.

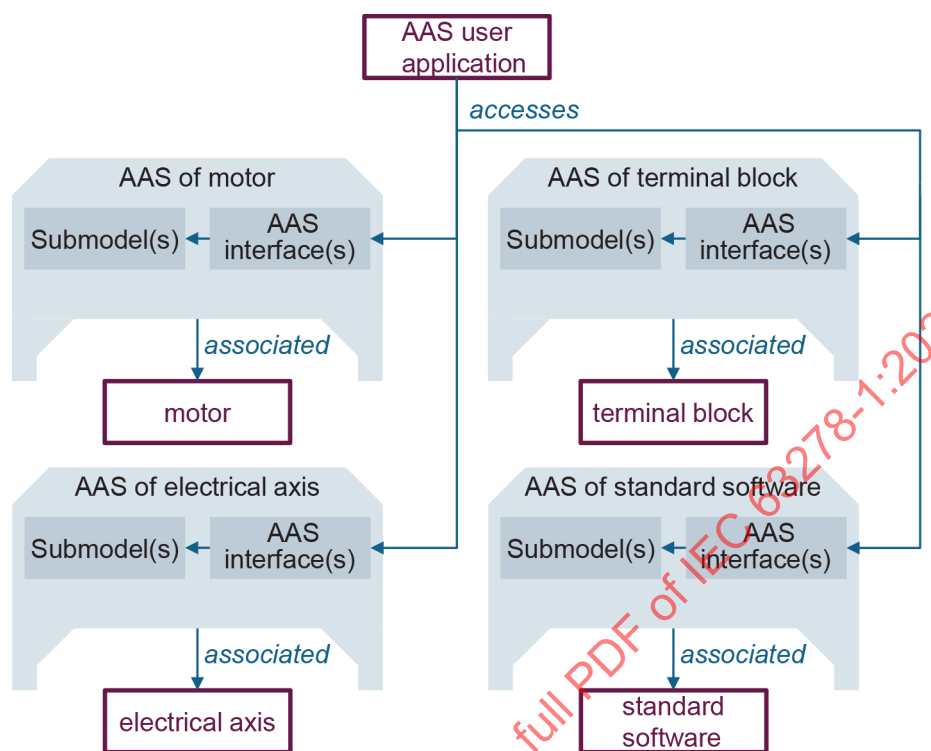
NOTE 3 The AAS interface(s) can be realized by an application programming interface (API). An Example of implementation of AAS interface(s) is OPC UA (see Clause A.7).

In industrial applications, multiple identifiers are used to retrieve information of an asset by software applications. Not all these identifiers are globally distinct but are defined by the context and are unique in a given namespace.

NOTE 4 Such an alternative identifier can be the global trade item identifier.

6.2.5.3 Example of the requirement

Subclause 6.2.5.3 and Figure 20 describe an example of the requirements.



IEC

EXAMPLE An Asset Administration Shell represents information and services associated with the respective asset. The idea of Asset Administration Shell makes it easier to maintain the digital representation congruent with the actual asset. In the example, for a set of assets, corresponding Asset Administration Shells are given.

To access these digital representations, a standardised interface is used. This interface can also be used for a lifelong maintenance of the information in the respective Asset Administration Shells and can provide efficient retrieval options. An AAS user application can access this interface (see 4.2.7).

Figure 20 – AAS user application accessing AAS interfaces of Asset Administration Shells

6.3 Requirements related to data exposure and information security

6.3.1 Data exposure of Asset Administration Shell and its Submodels

6.3.1.1 Requirements

An Asset Administration Shell shall support the control of the data exposure of the AAS management information and Submodels on an individual item basis in accordance with the AAS responsible's governance against each AAS user application.

6.3.1.2 Rationale

The Asset Administration Shell responsible does not want all of the information included in the Asset Administration Shell or Submodel, or both, to be exposed to all AAS user applications, but to control the exposure of that information in accordance with AAS responsible's business, personal or other strategies against each AAS user application.

Definition of appropriate security controls is one means to control the exposure of the Asset Administration Shell and its Submodels.

6.3.2 Requirements related to information security of Asset Administration Shell in general

6.3.2.1 Requirement

An Asset Administration Shell shall support the modelling of security propositions on availability, integrity and confidentiality for Submodels and AAS interface(s).

6.3.2.2 Rationale and guidance

The purpose of the Asset Administration Shell security is to protect the availability, integrity, and confidentiality of information and services provided by an Asset Administration Shell according to rules which are set by the AAS responsible. In addition to the SubmodelElements contained in the individual Submodels, the Asset Administration Shell shall provide means to provide information required for security governance.

Apart from the information and services provided by the Asset Administration Shell, Asset Administration Shell security shall also consider the security of the associated asset, AAS user application and Submodel template according to Figure 2. Especially, any negative impact of the Asset Administration Shell with regards to IT and OT security shall be inhibited.

Figure 21 shows an extension of Figure 2 regarding the Asset Administration Shell security aspects.

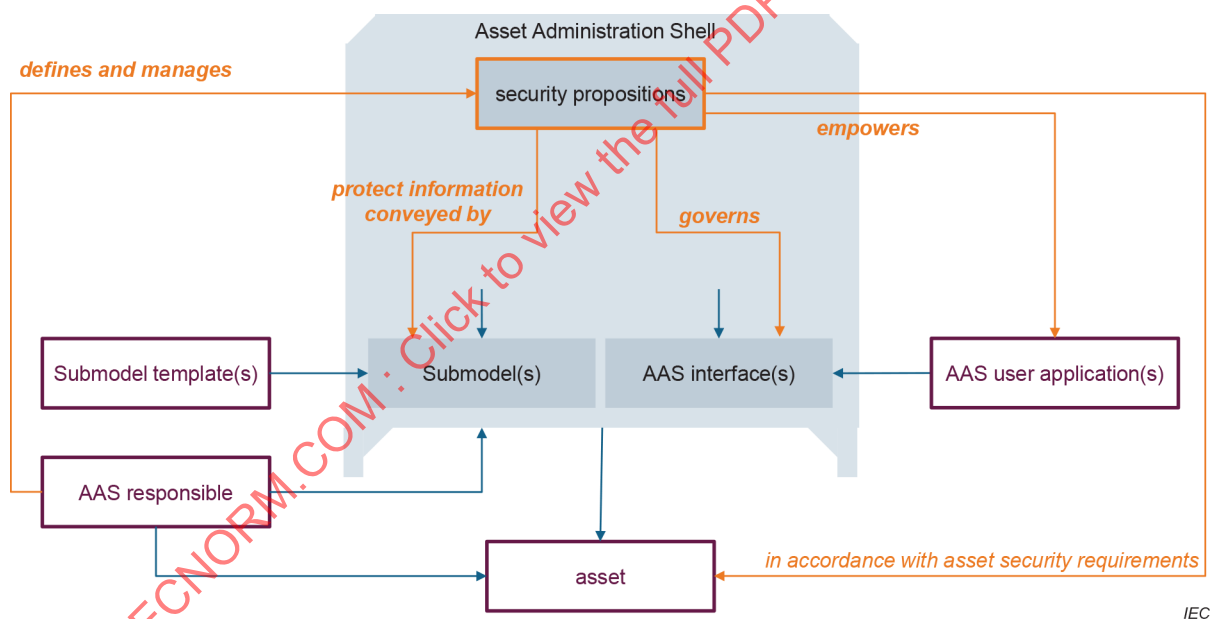


Figure 21 – Asset Administration Shell security overview

Security propositions describe how specific information and services provided by the Submodels are handled with respect to the protection goals' availability, integrity, and confidentiality. Security propositions may be realized as additional SubmodelElements of an existing Submodel or as separate dedicated security Submodels that relate to and augment existing Submodels.

The AAS responsible defines and manages the security propositions applicable to the information and services provided with or by the Submodels. Depending on a specific usage scenario a different set of qualities of availability, integrity and confidentiality can apply, e.g., the AAS responsible can decide which elements of a Submodel are made available to a specific AAS user application (confidentiality) and which provisions apply with respect to the integrity and availability (QoS) of the provided information and services. The AAS Interface controls the exposure of the information provided by the Submodels towards the AAS user application according to the security propositions.

Authorized AAS user applications are allowed to access the designated information and services and can rely on the information and services provided to the specified extend.

The AAS responsible shall also make sure that all information and services provided via an Asset Administration Shell are in accordance with the security requirements for the asset.

6.3.3 Requirements related to the IEC 62443 series

6.3.3.1 Requirement

An Asset Administration Shell shall support the implementation of IACS security in compliance with the IEC 62443 series.

6.3.3.2 Rationale and guidance

The IEC 62443 series is the prevalent series used to implement security for Industrial Automation and Control Systems (IACS). For security controls not directly addressed by the IEC 62443 series, the IEC 62443 series provides references to additional security standards such as the ISO/IEC 27000 series. The ISO/IEC 27000 series defines the implementation of an Information Security Management System (ISMS).

The structure of the Asset Administration Shell shall take this into account and shall be prepared to implement requirements resulting from the IEC 62443 series and ISO/IEC 27000 series.

Annex C provides some basics on the IEC 62443 series and its relation to the Asset Administration Shell (see C.1). The IEC 62443 series introduces foundational requirements (see C.2) and security levels (see C.3). The Asset Administration Shell shall reflect both which foundational requirements are affected, and which level of security applies to information and services provided by the Asset Administration Shell (compare to C.4).

Annex A (informative)

Relevant standards

A.1 Possible sources for Submodels and Submodel templates

A.1.1 General

Clause A.1 determines which sets of SubmodelElements are relevant depending on the use case. The source of the concept definition of SubmodelElements is needed to understand the semantics of the Submodel and its SubmodelElements. Also, it is important to consider in general cross-linking between SubmodelElements.

A.1.2 Different sets of concept definitions for SubmodelElements

A.1.2.1 General

An Asset Administration Shell is capable of holding information and services for different technical disciplines, different life cycle phases and different application and analysis scenarios. A large number and variety of concepts and classifications of products are applicable.

Concept repositories are the shared source for the semantic of SubmodelElements and specify concepts, property types, cardinalities, relationships between concepts and product classifications. This is done in the four following categories of publications (see Figure 7):

- standards;
- consortium specifications;
- manufacturer specifications;
- open community specifications.

These categories are characterized below.

A.1.2.2 Standards

Standards are developed and published by national or international standards developing organizations (SDOs) such as ISO, IEC, etc. Participation in the development of standards is open to any stakeholder. Access to the published standards is possible to any individual or organization.

The typical lead time for creating or amending a standard is three years, but accelerated procedures are available.

Standardized concept dictionaries specifying property types are published in different domains under the responsibility of the relevant technical committees of these SDOs. IEC provides the database for publishing semantic data structures (Common Data Dictionary "IEC CDD"). Refer to <https://cdd.iec.ch>.

The quality of the output is guaranteed by the participation of worldwide experts and a clearly defined process involving the choice of the stakeholders of participating national bodies.

The IEC CDD is a standard in database format managed by a change request procedure (based on ISO/IEC Directives Part 1 – IEC Supplement Annex SL), which reduces the duration of the procedure for importing and maintaining content (classes and property types) to 6 months.

A.1.2.3 Consortium specifications

Consortium specifications are developed and published by associations of stakeholders. Participation in a consortium and the development of consortium specifications requires in most cases a membership. Access to the publications is reserved to the members although some documents can be accessible freely.

For classes and property types, the corresponding organizations are called "product classification organization" such as ECLASS, ETIM, ECCMA, idea, JEITA ECALS dictionary.

The lead time for creating or amending a consortium specification depends on the consortium and the applied procedures. Due to a limited number of stakeholders the consensus can be quicker than for international standardization.

The coverage of published domains depends on the consortium.

A.1.2.4 Manufacturer specifications

Manufacturers own many private property types; more can be formed rapidly and can also address specific asset description.

Private classes and property types for specific assets that cannot be developed in conjunction with other stakeholders can be defined as part of a manufacturer dictionary and exposed by some product classification organization such as ECCMA.

A.1.2.5 Open community specifications

In an open community any entity or person can join and contribute to the collaborative effort without being a member of a certain consortium or organization. The resulting work is made available under a free license (e.g. Creative Commons). In addition, the data and the process are open to the public in all phases.

Open community specifications are developed in a process similar to open source software development. Such open community specifications define concepts which can be identified by an URI, e.g. "http://admin-shell.io/className/version/revision/propertyName".

EXAMPLE An example of an open community specification is
<https://github.com/admin-shell-io/tree/master/vdi/2770/1/0>.

A.1.3 Existing international specifications that can be used as basis for Submodel templates

International standards specify classes, features and information that can be portrayed in SubmodelElements and are therefore important for the concept of Asset Administration Shell (see also [1], ISO/IEC TR 63306-1 and -2, ISO TR 23087:2018). Each of these sources of SubmodelElements belong to a certain technical domain that can constitute a Submodel template and, by this, a referable context. However, standards and Submodel templates can be managed independently.

Table A.1 to Table A.3 show some examples:

- Table A.1 lists standards, which are directly providing definitions of concept repository entries, e.g. by means of IEC CDD.
- Table A.2 lists standards which provide potential sources of concept repository entries, which can be referenced by Submodel templates. These potential sources will require some refinement, before specific definitions of concept repository entries can be derived.
- Table A.3 lists standards which provide potential reference models, which are not solely defining concept repository entries, but also important overall concepts and definitions for syntactic interoperability.

**Table A.1 – Examples of standards providing concept repository entries
which can be referenced by Submodel templates**

Specification / standard		Technical domain
Number	Title	
IEC 62683 (all parts)	Low-voltage switchgear and controlgear – Product data and properties for information exchange	Low-voltage switchgear and controlgear
IEC 61987 (all parts)	Industrial-process measurement and control – Data structures and elements in process equipment catalogues	Process control field devices
ISO 13399 (all parts)	Cutting tool data representation and exchange	Cutting tools
ISO 13584-42	Industrial automation systems and integration – Parts library – Part 42: Description methodology: Methodology for structuring parts families	Parts families
ISO 13584-501	Industrial automation systems and integration – Parts library – Part 501: Reference dictionary for measuring instruments – Registration procedure	Measuring instruments
ISO 13584-511	Industrial automation systems and integration – Parts library – Part 511: Mechanical systems and components for general use – Reference dictionary for fasteners	Mechanical systems and components
ISO/TS 15926-4	Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities – Part 4: Initial reference data	Process plants including oil and gas production
ISO 18582 (all parts)	Fluid power – Specification of reference dictionary	Fluid power components
ISO 23584 (all parts)	Optics and photonics – Specification of reference dictionary	Optics and photonics
ISO 25297 (all parts)	Optics and photonics – Electronic exchange of optical data	Optics and photonics

Table A.2 – Examples of standards providing potential sources of Submodel templates

Specification / standard		Technical domain
Number	Title	
IEC 61784 (all parts)	Industrial communication networks – Profiles	Industrial network
ISO 12100	Safety of machinery – General principles for design – Risk assessment and risk reduction	Safety
ISO 13849-1	Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design	
IEC 62061	Safety of machinery – Functional safety of safety-related control systems	
IEC 61511 (all parts)	Functional safety – Safety instrumented systems for the process industry sector	
IEC 61508 (all parts)	Functional safety of electrical/electronic/programmable electronic safety-related systems	
IEC 62264-2 and IEC 62264-4	Enterprise-control system integration – Part 2: Object and attributes for enterprise-control system integration – Part 4: Object model attributes for manufacturing operations management integration	Enterprise-control system integration
IEC 62443 (all parts)	Industrial communication networks – Network and system security	Security
ISO 20140-5	Automation systems and integration – Evaluating energy efficiency and other factors of manufacturing systems that influence the environment – Part 5: Environmental performance evaluation data	Energy efficiency
IEC 61915 (all parts)	Low-voltage switchgear and controlgear – Device profiles for networked industrial devices	Low-voltage switchgear and controlgear
IEC 62271 (all parts)	High-voltage switchgear and controlgear	High-voltage switchgear and controlgear
IEC TR 63213	Power measurement applications within electrical distribution networks and electrical installations (including database definitions)	Measuring equipment domain for electrical and electromagnetic quantities
IEC 61800-7 (all parts)	Adjustable speed electrical power drive systems – Part 7-X: Generic interface and use of profiles for power drive systems	Motion & drives
IEC TS 62872-1	Industrial-process measurement, control and automation – Part1: System interface between industrial facilities and the smart grid	Energy efficiency
IEC 62872-2	Industrial-process measurement, control and automation Part 2: Internet of Things (IoT)– Application framework for industrial facility demand response energy management	
ISO 10303 (all parts)	Industrial automation systems and integration — Product data representation and exchange	Manufactured products
ISO 22400-1 and ISO 22400-2	Automation systems and integration – Key performance indicators (KPIs) for manufacturing operations management – Part 1: Overview, concepts and terminology – Part 2: Definitions and descriptions	Key performance indicators (KPIs) for manufacturing operations management

Table A.3 – Examples of standards providing reference models for Submodels

Specification / standard		Technical domain
Number	Title	
IEC 62714 (all parts)	Engineering data exchange format for use in industrial automation systems engineering – Automation markup language	Overall (Asset Administration Shell)
IEC 62541 (all parts)	OPC unified architecture	
IEC 62832 (all parts)	Industrial-process measurement, control and automation – Digital factory framework	
IEC 61360 (all parts)	Standard data element types with associated classification scheme for electric items	
ISO 13584-42	Industrial automation systems and integration – Parts library – Part 42: Description methodology: Methodology for structuring parts families	
ISO/IEC 6523	Information technology – Structure for the identification of organizations and organization parts	Identification
ISO/IEC 11179 (all parts)	Information technology – Metadata registries (MDR)	
ISO TS 29002-5	Industrial automation systems and integration – Exchange of characteristic data – Part 5: Identification scheme	
IEC 61804 (all parts)	Devices and integration in enterprise systems – Function blocks (FB) for process control and electronic device description language (EDDL)	Configuration
IEC 62769 (all parts)	Field device integration (FDI)	
IEC 62453 (all parts)	Field device tool (FDT) interface specification	
ISO 15745 (all parts)	Industrial automation systems and integration – Open systems application integration framework	
IEC 62890	Industrial-process measurement, control and automation – Life-cycle management for systems and components	Life cycle
IEC TS 62720	Identification of units of measurement for computer-based processing	Units of measurement
ISO 16100-1	Industrial automation systems and integration – Manufacturing software capability profiling for interoperability – Part 1: Framework	Manufacturing software capability profiling
ISO 16400-1	Automation systems and integration – Equipment behaviour catalogues for virtual production system – Part 1: Overview	Equipment behaviour

A.2 IEC 61360 dictionaries, classes and property types

A.2.1 General

The IEC 61360 series provides a framework and an information model for product dictionaries.

The concept of product type is represented by "classes" and the product characteristics are represented by "properties".

A.2.2 Classes

A hierarchy of classes is a structure of classes according to the commonalities of the properties in the classes ("taxonomy"). Properties associated with classes can be inherited. Figure A.1 shows an example.

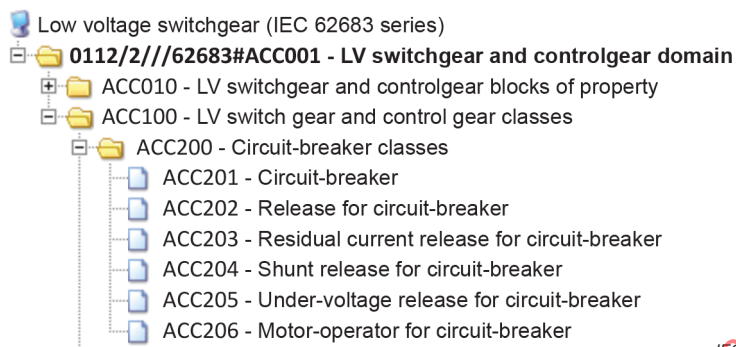


Figure A.1 – Example: Representation of a class tree in IEC CDD

A.2.3 Property types and instances

A property instance, when used, has an associated value, such as a numerical value together with a unit, or a qualitative value taken from a predefined list.

The information model divides the attributes of property types (see Table A.4 for examples) into four main groups:

- 1) identifying attributes (e.g. code, version number, revision number, preferred name, synonymous name, short name),
- 2) semantic attributes (e.g. definition, note, remark, source document of definition),
- 3) value attributes (data type, value format, unit of measure, alternative unit of measure, references class identifier, number of significant digits) and
- 4) relationship attributes.

Table A.4 – Example of representation of a property type with some attributes in IEC CDD

Code:	0112/2///62683#ACE424
Version:	001
Revision:	01
IRDI:	0112/2///62683#ACE424#001
Preferred name:	rated current
Synonymous name:	
Symbol:	In
Synonymous symbol:	
Short name:	
Definition:	maximum uninterrupted current equal to the conventional free-air thermal current (Ith)
Note:	
Remark:	
Primary unit:	A
Alternative units:	
Level:	
Data type:	LEVEL(MAX) OF REAL_MEASURE_TYPE

IEC

A.2.4 IEC Common Data Dictionary (IEC CDD)

Based on the IEC 61360 series, classes and property dictionaries (domain ontologies) for the description of production resources can be created.

IEC CDD – based on the IEC 61360 series – can be used for templates or structures for defining company specific data collections for various applications, such as catalogues, databases, and master-data repositories. Such consistent data collections are indispensable for data integrity and efficient management of corporate product and service portfolios including data sharing.

IEC Common Data Dictionary (IEC CDD), which is a standard in database format, is a common repository of concepts for all electrotechnical domains based on the methodology and the information model of the IEC 61360 series, and provides:

- unambiguous identification of classes and properties, and their relations;
- commonly accepted terminology and definitions based on accepted sources such as IEC International Standards, other International Standards, industry standards, or public authorities;
- hierarchies of concepts enabling users to appropriately characterize their products and services;
- relevant conditions and constraints if necessary on possible values of characteristics;
- technical representation of concepts including units and data types and their identification.

IEC CDD is structured into separate dictionaries (named domains) for different technical areas. Each dictionary defines classes and properties based on one or more international standard. Currently four dictionaries are accessible:

- 1) electric/electronic components (IEC 61360-4)
- 2) process automation (IEC 61987 series) (see Clause A.3)
- 3) low voltage switchgear (IEC 62683 series) (see Clause A.4)
- 4) measuring equipment for electrical quantities (IEC TR 63213)

Additionally, IEC CDD provides a dictionary with nearly 2 000 units of measurement. Each unit of measurement is provided with an IRDI and a definition based on IEC TS 62720 and with reference to UN/ECE Recommendation 20 [5].

A dictionary of IEC CDD can reference content from other dictionaries. For instance, all four dictionaries listed above are referencing the units of measurement from the IEC TS 62720 dictionary, thus using a common base for providing measured values.

A.2.5 ECLASS

ECLASS – based on ISO 13584-42 (equivalent to IEC 61360-2) – is a concept repository provided by the ECLASS consortia.

The ECLASS dictionary provides a product classification and product description for a wide range of technical and other domains (e.g. electrical engineering, logistics, optics, medical) enabling global collaboration through the exchange of standardized master data.

Similar to IEC CDD it provides:

- unambiguous identification of classes and properties, and their relations;
- commonly accepted terminology and definitions;
- classification of concepts enabling users to appropriately characterize their products and services;
- relevant conditions and constraints, if necessary, on possible values of characteristics;
- technical representation of concepts including units and data types and their identification.

A.3 IEC 61987 series classes and dictionary

A.3.1 General

The IEC 61987 series provides a product data framework based on the IEC 61360 series, and based on that framework defines product data for devices used in the process industry.

The IEC 61987 series framework is based on specific classes such as:

- a) properties (as defined by the IEC 61360 series)
- b) list of properties (LOP)
 - 1) DLOP (device list of properties)
 - 2) OLOP (operating list of properties)
 - 3) additional aspects
- c) blocks of properties
- d) views

IEC 61987-10 provides definitions for these classes and introduces additional concepts like cardinality of blocks, polymorphism, composition and aggregation.

These definitions are provided in IEC 61987-10. Subsequent parts of the IEC 61987 series define product data representations for devices used in the process industry (e.g. IEC 61987-12 defines LOPs for flow measuring devices, IEC 61987-32² defines LOPs for I/O modules, IEC 61987-92 defines aspect LOPs).

² Under development. Stage at the time of publication: IEC CDV 61987-32:2022,

A.3.2 Specific classes

a) List of properties

A list of properties (LOP), which is a subtype of the IEC 61360 series item class, is a compilation of properties. Such a list can be "linear" (not structured) or structured. A structured LOP uses blocks of properties to represent a feature or an aspect of an object.

A list of properties is intended to represent a device. Different aspects of the device are represented by different types of LOPs (see Figure A.2):

- the device LOP (DLOP) represents the characteristics of the device;
- the operating LOP (OLOP) represents aspects of the operational environment of the device;
- the administrative LOP (ALOP) represents administrative information for execution of business transactions;
- the commercial LOP (CLOP) represents commercial information related to business transactions.

The IEC 61987 series distinguishes especially between DLOP (device list of properties) and OLOP (operating list of properties) which can have different aspects concerning the necessary compilation.

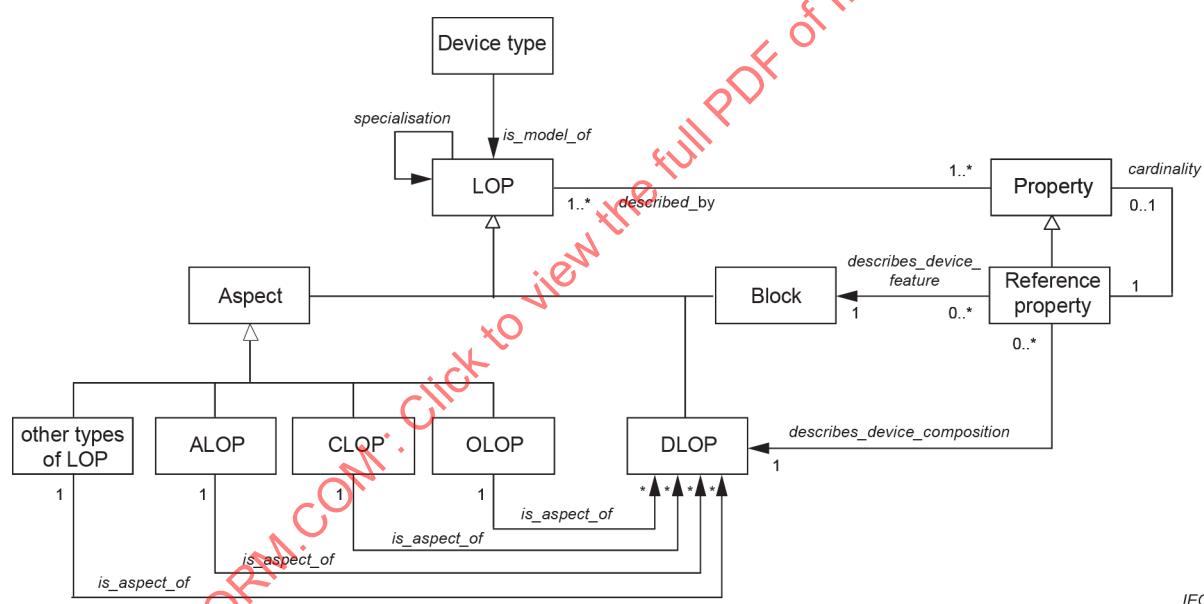


Figure A.2 – Simplified UML scheme of device, LOPs and aspects (see IEC 61987-11)

b) Blocks of properties

A block of properties (block), which is a subtype of a LOP, consists of one or more properties describing an abstraction of a feature of a device type.

Blocks can be used to provide a hierarchical organization of properties.

The organization of the blocks can be functionally oriented.

c) Views

According to IEC 61987-10 a view defines the specific set of properties required for a specific transaction between business partners. A view allows to select only the data required for executing the transaction by setting or cancelling a filter for properties or block of properties.

A.3.3 Dictionary

The dictionary for the IEC 61987 series is hosted at the IEC CDD (see A.1.2 and A.1.3). The dictionary is represented in the graphical user interface of the IEC CDD as the domain “Process automation (IEC 61987 series)”.

The IEC 61987 series dictionary references concepts from other dictionaries (e.g. unit definitions from the IEC TS 62720 dictionary and value lists from the IEC 61360 series dictionary).

A.4 IEC 62683 series classes and dictionary

A.4.1 General

The IEC 62683 series provides a product data framework based on the IEC 61360 series, and based on that framework defines product data for low-voltage switchgear and controlgear devices used for example in the factory automation industry.

The IEC 62683 series framework is based on specific classes such as:

-
-  Low voltage switchgear
 -   0112/2///62683#ACC001 – –V switchgear and controlgear domain
 -    ACC002 – E-gineering data for low voltage switchgear, controlgear and their assemblies
 -    ACC010 – L– switchgear and controlgear blocks of properties
 -   ACC100 – L– switchgear and controlgear classes

The IEC 62683 series uses the concepts defined in the IEC 61987 series, such as cardinality of blocks, polymorphism, composition and aggregation.

For example the IEC 62683 series for the class "Circuit-breaker" defines the following blocks (see Figure A.3):

- Identification
- General technical data
- Main circuit
- Short-circuit
- Over-current release
- Control and auxiliary circuits
- Data communication
- Installation, mounting and dimensions
- Connection facilities
- Product certificates and standards

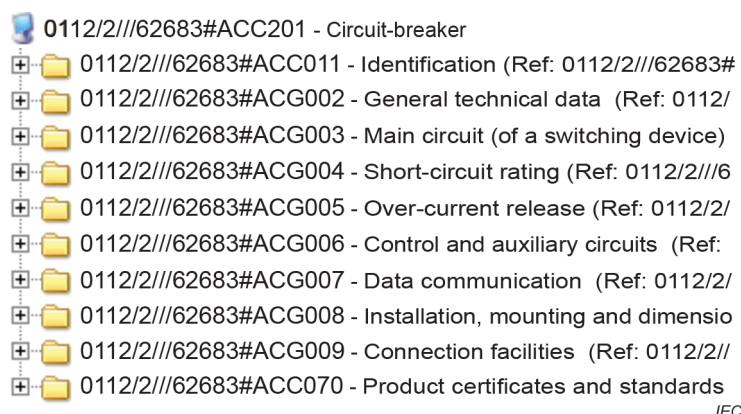


Figure A.3 – Organization in blocks

General definitions are provided in IEC 62683-1 for product data for information exchange (e.g. catalogue data). Engineering data aspects are considered in IEC 62683 series for engineering data for computer-aided design and manufacturing of assemblies and in the IEC 62683 series for switchgear and controlgear assembly objects for BIM (Building Information Modelling).

A.4.2 Dictionary

The dictionary for the IEC 62683 series is hosted at the IEC CDD (see A.1.2 and A.1.3). The dictionary is represented in the GUI of the IEC CDD as the domain "Low voltage switchgear and controlgear" (IEC 62683 series).

The IEC 62683 series dictionary references concepts from other dictionaries (e.g. unit definitions from the IEC TS 62720 dictionary and value lists from the IEC 61360 series dictionary).

A.5 Digital Factory (IEC 62832 series)

A.5.1 Introduction to Digital Factory

The standard for the Digital Factory, IEC 62832-1, defines a framework of abstract definitions for:

- assets of production systems,
- structural and behavioural relationships,
- feature (property) management,
- hierarchical relationships,
- technical aspects.

An asset is understood as a physical or logical object which is owned or managed by an organization and which has an actual or perceived value for the organization. IEC 62832-1 addresses in particular the utilization phase of such assets and the related production facilities, their design, construction, commissioning, operation and maintenance. For the identification of concepts, IEC 62832-1 uses identifiers in accordance with ISO TS 29002-5. The assets under consideration can also be identified by means of other identifiers (e.g. URI).

Assets (real or logical items) are described by means of asset descriptions ("DF asset", digital representation). Classes of assets are modelled by means of asset classes and thus stand for one or more assets which share the same set of features (see Figure A.4). The features of the assets are described by means of data elements. If the described assets have a modular structure, the corresponding asset descriptions (asset classes) can similarly describe a modular structure. In this case the asset class describes the modules as assets and represents the links between the assets.

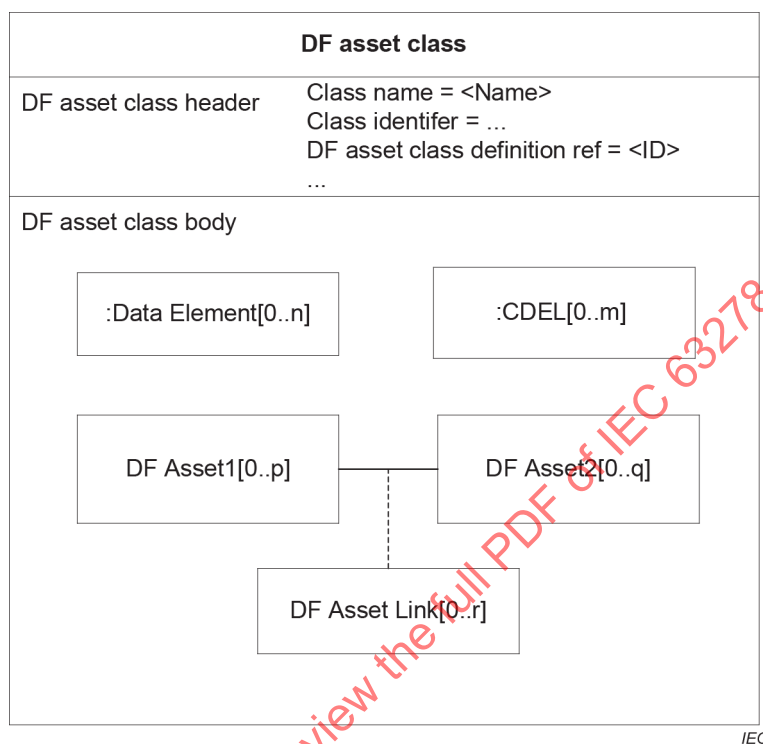


Figure A.4 – Structured asset class in the Digital Factory (see IEC 62832 series)

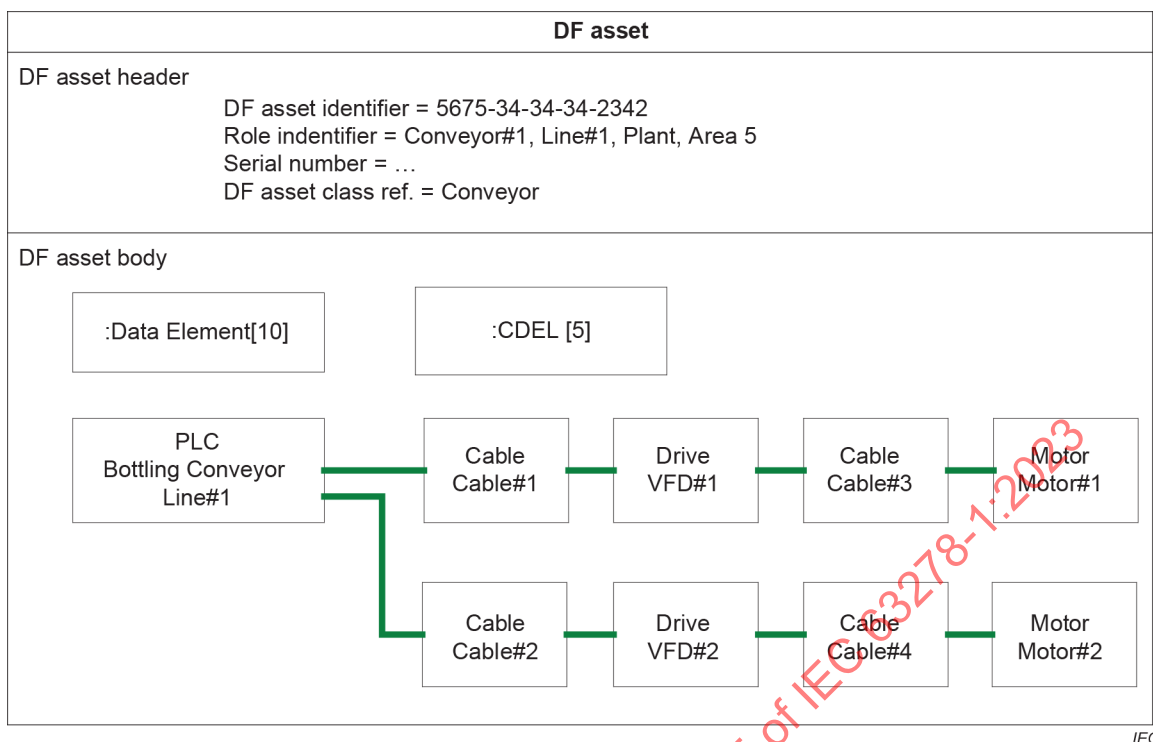


Figure A.5 – Description of a structured asset that is composed of several assets

Most important for the description of an asset is the breakdown of the description into "header" and "body". The "header" contains information for identification and designation of the specific asset in the respective production system and stipulates aspects for administration of the asset. The "body" contains data for description of the features of the asset class with their respective disposition (i.e. data element values) for the specific asset. For the specification of data elements, the Digital Factory makes reference to the IEC 61360 series properties (see Figure A.5) and the ISO 22745 series.

The main contribution of the Digital Factory framework is the clear and unambiguous separation of data domains and their respective concepts:

- Dictionary domain with dictionaries for providing DF asset class definitions (i.e. templates for asset descriptions).
- Library domain with libraries for providing DF asset classes (i.e. type asset descriptions).
- Digital Factory domain providing DF assets (i.e. description of the real or logical assets).

This separation allows to define clear rules within the different domains (e.g. regarding the usage of identifier types), allows for automated processing of the data and enables the usage of property-based data for engineering use cases (e.g. for supporting design decisions based on library data or for support of design validation based on Digital Factory data).

These concepts can be applied in conjunction with other design languages (e.g. Automation ML and SysML).

A.5.2 Compatibility of the Digital Factory with the concept of Asset Administration Shell

The objective of the work in the IEC is to create a standard (known as "Digital Factory") for electronic, machine-readable representations of production systems which can be used in all life cycle phases of the production systems. At the same time, exchange of information between the different participants in the design, construction and operation of a production system is intended. An important area of focus is the semantic clarity of the information, which is why the standard "Digital Factory" is based on property descriptions of the components of the production system. The utilisation of property descriptions for the components and component groups allows automated processing of the description of the production system, e. g. in the validation of a system design. The standard "Digital Factory" makes no assumption regarding the management of the data in the description of the production system (i.e. no central repository is defined).

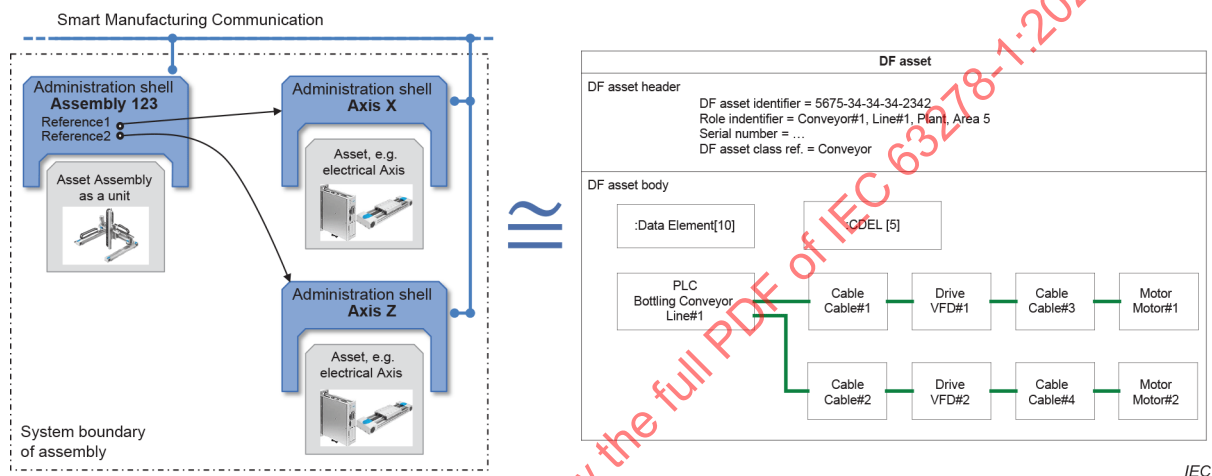


Figure A.6 – Comparison of the approaches of the Digital Factory and Asset Administration Shell

An objective of smart manufacturing is automated configuration of a production system in accordance with varying production orders. This objective can only be achieved based on a machine-processible description of the production system and the products to be manufactured. The information model for the Asset Administration Shell is therefore based on the standard "Digital Factory" and extends it to include further required information (see Figure A.6). In order to achieve the required flexibility for the automated configuration of a production system the production resources are not only represented by passive data elements, but also by independent Asset Administration Shells with services and API, which are not only centrally administered but can also be understood as being a distributed repository. The data elements of the Digital Factory (referred to here as DF assets) at the same time form the basis for the Asset Administration Shells.

Table A.5 gives a comparison of the individual concepts of Digital Factory (IEC 62832 series) and Asset Administration Shell. Table A.5 is not concerned about the existence of services of the Asset Administration Shells or APIs. Such concepts are not in the scope of the IEC 62832 series.

Table A.5 – Comparison of the individual concepts of Digital Factory (IEC 62832 series) and Asset Administration Shell

Domain of description	Concept	DF model element	Asset Administration Shell (subclause)
Meta type domain	Template for representing information associated with a domain and with distinguishable life cycle	Special type of CDEL definition	Submodel template (4.2.10)
	Template for representing a characteristic of a component	DF data element type	Property type (3.1.21)
Type asset domain	Representation of simple classes of components for production system	DF asset class	Asset Administration Shell (type asset) (4.2.3)
	Representation of composite component class for production system	Composite DF asset class	Composite Asset Administration Shell (Figure 19)
	Identification or management of the representation	DF asset class header	Management information of the Asset Administration Shell (type asset) (6.2.2.1)
	Identification of the represented type asset (class of component)	<to be defined by the implementation specification>	ID of the asset (6.2.2.1)
	Represented information on the type asset	DF asset class body	Set of Submodels of Asset Administration Shell (type asset) (4.2.8)
	Hierarchical structuring of properties	CDEL	Collections of SubmodelElement (4.2.9)
	Information associated with a domain and with distinguishable life cycle	Special type of CDEL	Submodel (Asset Administration Shell with type asset) (4.2.8)
	Representation of a characteristic of component class	Data element	Property (property instance), SubmodelElements (4.2.9)
Instance asset domain	Representation of simple component of production system	Physical asset information of DF asset	Asset Administration Shell (instance asset) (4.2.3)
	Representation of a composite component of production system	Composite DF asset	Composite Asset Administration Shell (Figure 19)
	Identification/management of the representation	DF asset header	Management information of the Asset Administration Shell (instance asset) (6.2.2.1)
	Identification of the represented asset	PS asset identifier	ID of the asset (6.2.2.1)
	Represented information on the instance asset	DF asset body	Set of Submodels of Asset Administration Shell (instance asset) (4.2.8)
	Hierarchical structuring of properties	CDEL	Collections of SubmodelElements (4.2.9)
	Information associated with a domain and with distinguishable life cycle	Special type of CDEL	Submodel (Asset Administration Shell with instance asset) (4.2.8)
	Representation of a characteristic of component or of production system	Data element	Property (property instance) SubmodelElements (4.2.9)
Real world	Physical or logical object owned by or under the custodial duties of an organization, having either a perceived or actual value to the organization	Asset	Asset (4.2.3)
	Asset which is part of the production system	PS asset	Asset (4.2.3)

NOTE 1 An Asset Administration Shell has additional elements, for instance an AAS interface.

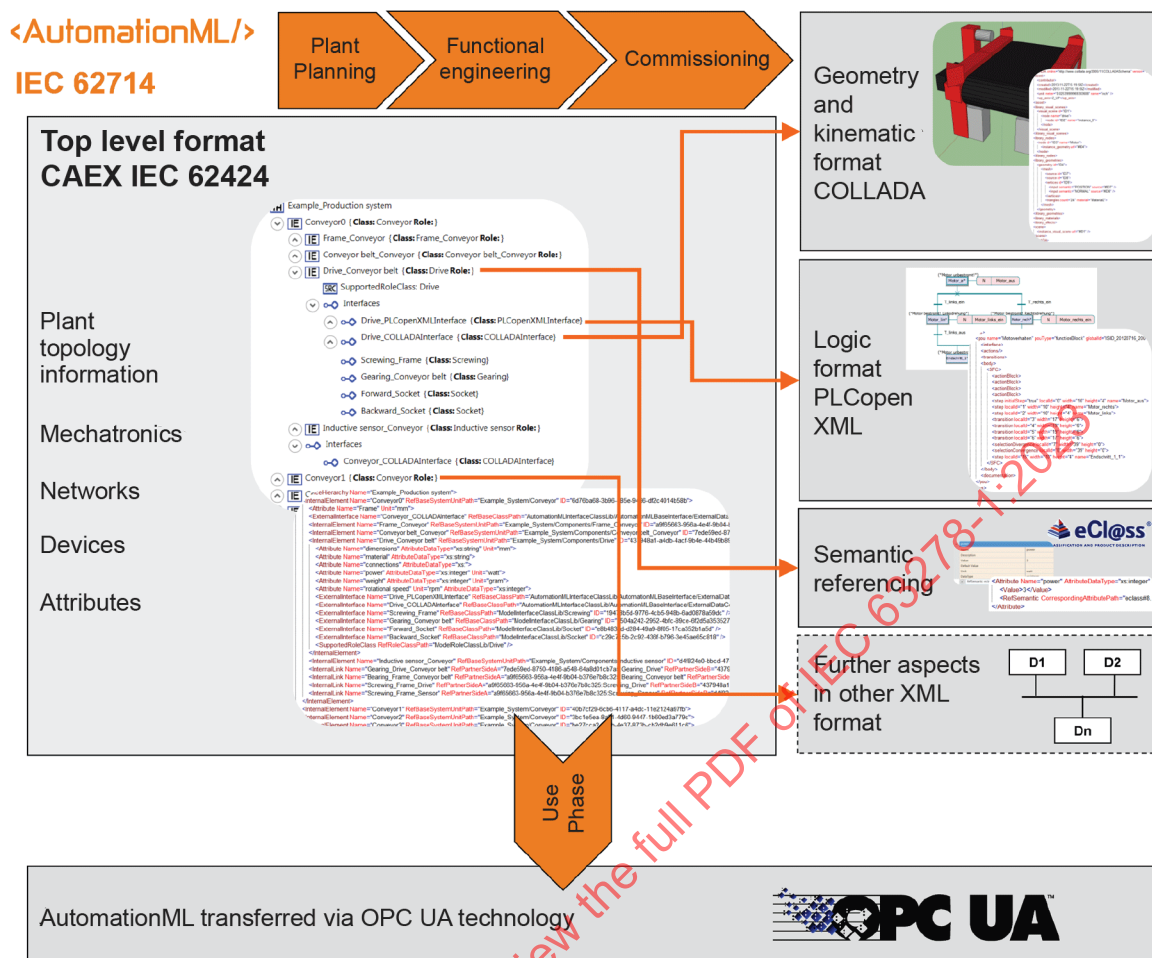
NOTE 2 There are additional purposes of SubmodelElements, for instance service.

This document specifies that Submodels can be based on dictionaries but also on ontologies (see 4.2.12). Since a Submodel is defined for a specific use case and can be based on either dictionaries or ontologies, the rules of the IEC 62832 series apply only to some Submodels (which are based on dictionaries) and only for specific use cases (e.g. systems engineering).

A.6 AutomationML (IEC 62714 series)

A.6.1 AutomationML overview

AutomationML (AML) is a scalable, open, neutral, XML based, extendable, and free data exchange format, standardized in the IEC 62714 series. It has been developed in order to support the data exchange in a heterogeneous engineering tools landscape by enabling a discipline and company crossing transfer of engineering data between software tools – in a consistent manner and without data loss. With the meta-model provided by AML, production system structures can be modelled and expressed as a hierarchy of AutomationML objects. They can contain geometry and kinematics, logic data (behaviour and sequencing), and relations among AutomationML objects as well as references to information that is stored in external documents. Following this object-oriented paradigm, each AutomationML object can integrate different information with different semantics related to different disciplines. Figure A.7 provides an overview of the distributed document architecture of AML.



IEC

Figure A.7 – Architecture of AutomationML

CAEX is the top level format of AML and is characterized by the four following concepts (see Figure A.8), where the AML objects are represented as InternalElements (IE):

- By using RoleClasses (RC), semantics for AML objects can be modeled. RoleClasses describe functions of a physical or logical object independent of a technical implementation. They offer a possibility to specify an object in an abstract way and independent of the manufacturer. They are stored in RoleClass libraries.
- By using InterfaceClasses (IC), the semantics for the interfaces of AML objects can be modeled. InterfaceClasses describe relations of physical or logical objects or references to documents independent of a technical implementation. They are stored in interface class libraries.
- By using SystemUnitClasses (SUC), types or templates of AML objects can be modeled. They are stored in system unit class libraries.
- The InstanceHierarchy (IH) stores the actual project.

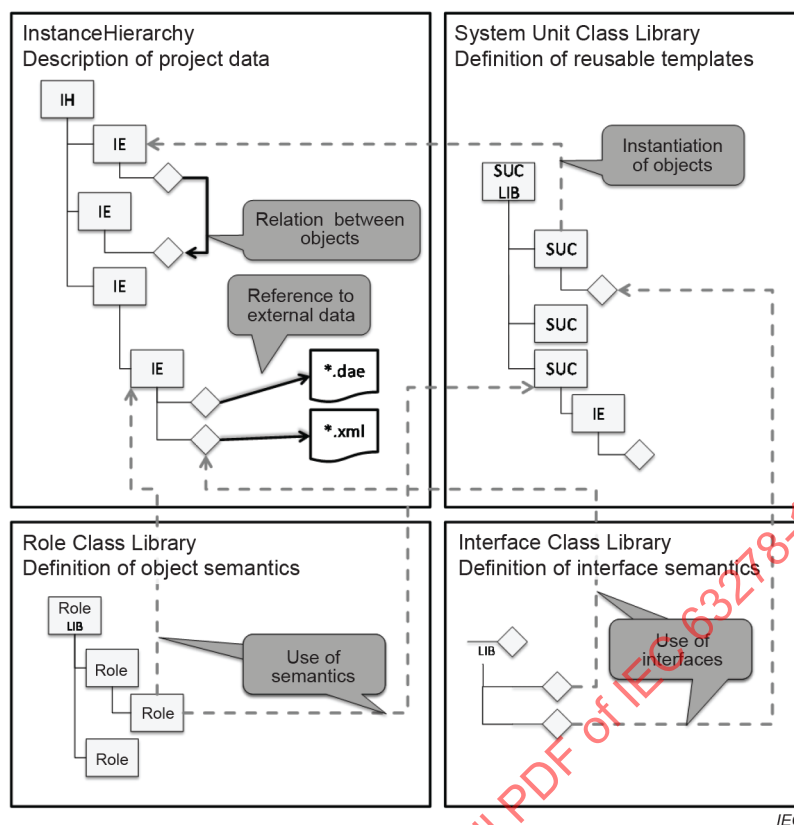


Figure A.8 – CAEX concepts

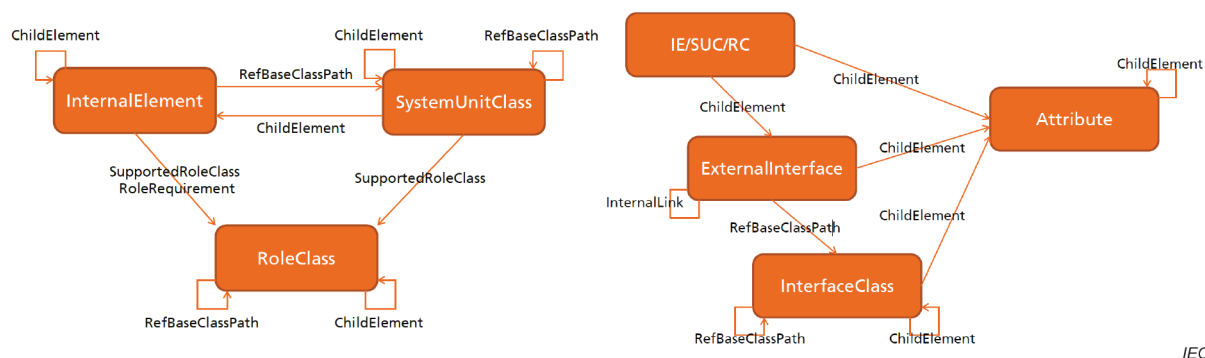
For a consistent data exchange, an object identification concept is essential. To support this concept, AML provides and specifies that every object will have an UUID (Universally Unique Identifier), which is kept as long as the object exists. With this ID persistence it is possible to identify a certain AML object throughout its entire life cycle, including archiving.

The AML mechanism to reference external documents enables the integration of further information aspects to AML objects. Therefore, it provides a special set of interface classes and role classes being used to uniquely identify the document type and the semantics of attached information. In case the referenced document has a unique identification concept for its content, it is also possible to reference inside the referenced document.

Besides the AML meta-model, there is the specification for the data exchange format available as well. Since AML is an XML based format, XML specific features are available, like XML encryption, XML signature, or XML serialization.

A.6.2 AutomationML modeling concepts

The main AutomationML elements and all possible relations between them are shown in Figure A.9. The organization in a hierarchical tree involves that every element type can have child elements of the same type (see ChildElement relations). SystemUnitClasses and RoleClasses are defined in an inheritance structure; the RefBaseClassPath relations define such inheritance relations.



IEC

Figure A.9 – Relations between AutomationML elements

An InternalElement is inherited from a role class and can inherit from a SystemUnitClass. In the scope of AML, SystemUnitClasses are only templates and can be changed after instantiation. This allows a modification of the AML object regarding instance specific needs, e.g. deleting, editing, or adding attributes and interfaces. However, it is also possible to model AML objects directly into the InstanceHierarchy – without having instantiated a SystemUnitClass. The backward relation indicates that a SystemUnitClass can define sub-elements in terms of InternalElements. These sub-elements (InternalElements) can be included on instantiation. RoleClass, SystemUnitClass, or InternalElement can contain arbitrary nested Attributes and Interfaces (see ChildElement relations). The Interfaces are called ExternalInterfaces and inherit from an InterfaceClass. They can be connected to other ExternalElements via InternalLinks. The InterfaceClasses are stored hierarchically and have independent inheritance relations. InterfaceClasses and certainly ExternalInterfaces can also have arbitrary nested attributes.

RoleClasses can be assigned to SystemUnitClasses and InternalElements. In the latter case there are two different ways to make the RoleClass assignment, using SupportedRoleClasses or the RoleRequirement. The remaining arrows in the left diagram of Figure A.9 describe the possible relations between InternalElements and SystemUnitClasses.

When RoleClasses are assigned to SystemUnitClasses or InternalElements or when InterfaceClasses are assigned to ExternalInterfaces, their attributes can be copied, manipulated, or deleted. No strict inheritance relation is applied here, either.

A.6.3 Interoperability of Asset Administration Shell supported by AutomationML

With respect to the structure of the Asset Administration Shell, AML provides a number of concepts that comply with the concepts of the structure of the Asset Administration Shell as well as with many of the relevant standards described in Annex A. Due to the strong focus of AML in the field of manufacturing engineering, many engineering tools that support AML can already provide structured information of assets, e.g. automation system components. These can be directly consumed for the descriptive part of the header and the individual sub models in the body of the Asset Administration Shell.

Concerning the different life cycle phases that need to be supported from the Asset Administration Shell, AML further complies with this concept with the following AutomationML main elements:

- RoleClasses, which represents semantics of objects (e.g. parallel gripper);
- SystemUnitClasses, representing object types, which can be linked to role classes (e.g. specific product family of parallel gripper from a certain manufacturer);
- InternalElements that represent the object (type) instances, which can be linked to role classes or System Unit Classes, or both (Actual parallel gripper with an individual serial number);

- InterfaceClasses that allow semantic relations between the AutomationML objects or references to documents (e.g. mechanical connection between parallel gripper with robot or reference to 3D-CAD representation).

Though AML follows the object-oriented paradigm, it is not as strict as it is defined in object-oriented programming. This allows a more flexible representation of instances in a data model, by changing the attributes of an InternalElement independent from the SystemUnitClass. AML even allows to represent inconsistent data models by design, and this addresses the real life situation of incomplete models during the first life cycle phases.

The standardized meta-model of AML makes no assumptions on the actual application domain. Hence, it can be used to represent the self-sufficient description of arbitrary data structures for the header of the Asset Administration Shell. Further, it can be used for the individual description of the sub-model data structures representing the possible different aspects of the asset to be represented in the body of the Asset Administration Shell.

This self-sufficient describing capability through AML is an important pre-requisite for realizing interoperability between arbitrary types of assets (physical, logical, software, etc.) represented by an Asset Administration Shell.

A.7 OPC UA

A.7.1 OPC UA overview

A.7.1.1 General

OPC Unified Architecture (UA) is a series of standards that support the vertical and horizontal connectivity between components, applications, and functions. OPC UA is applicable to components in all industrial domains, such as industrial sensors and actuators, control systems, manufacturing execution systems (MES) and enterprise resource planning (ERP) systems, etc. OPC UA defines a common infrastructure model to facilitate an information exchange. OPC UA specifies the following:

- the information model to represent structure, behaviour and semantics;
- the interface model to interact between applications;
- the communication model to transfer the data between end-points;
- the conformance model to guarantee interoperability between systems.

The OPC UA series is organized as a multi-part specification, as illustrated in Figure A.10.

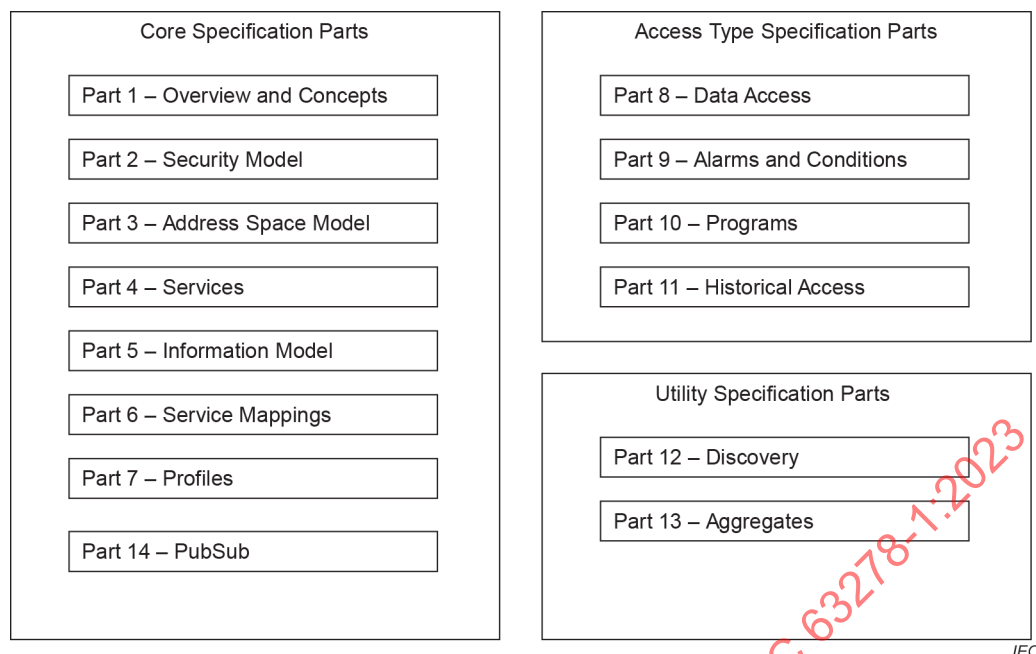


Figure A.10 – OPC UA specification organization

A.7.1.2 Secure, reliable and platform independent exchange of information

OPC UA is the latest technology initiated and maintained within the OPC Foundation for the secure, reliable and vendor-independent transport of raw data and pre-processed information from sensor and field level, through the control system level up to and into enterprise level with e.g. ERP, MOM, and PLM systems. With OPC UA every type of information is available anytime and anywhere for every authorized use and to every authorized person.

A.7.1.3 Platform and vendor-independent

OPC UA is independent of the vendor or system supplier that produces or supplies the respective application. The communication is independent of the programming language in which the respective software was programmed and it is independent of the operating system on which the application runs. It is an open standard without any dependence on, or binding to, proprietary technologies or individual vendors.

A.7.1.4 Standardized communication via Internet and firewalls

OPC UA extends the preceding OPC industry standard with several important functions such as platform independence, scalability, high availability and Internet capability. OPC UA is very simple to adapt. It connects the enterprise level down to the embedded systems of the automation components, independently of operating systems. OPC UA uses TCP for an optimized, binary protocol. Web service and HTTP are optionally supported. Additional protocol bindings can be integrated easily. The integrated encryption mechanisms ensure secure communication over the Internet.

A.7.1.5 Service-oriented architecture

OPC UA defines generic services and in doing so follows the design paradigm of service-oriented architecture (SOA), with which a service provider receives requests, processes them, and sends the results back with the response.

A.7.1.6 Protection against unauthorized access

The OPC UA technology uses proven security concepts that offer protection against unauthorized access, against sabotage, against the modification of data and against careless operation. The OPC UA security concepts contain user and application authentication, the signing of messages and the encryption of the transmitted data itself. OPC UA security is based on recognized standards that are also used for secure communication on the Internet, such as SSL, TLS and AES.

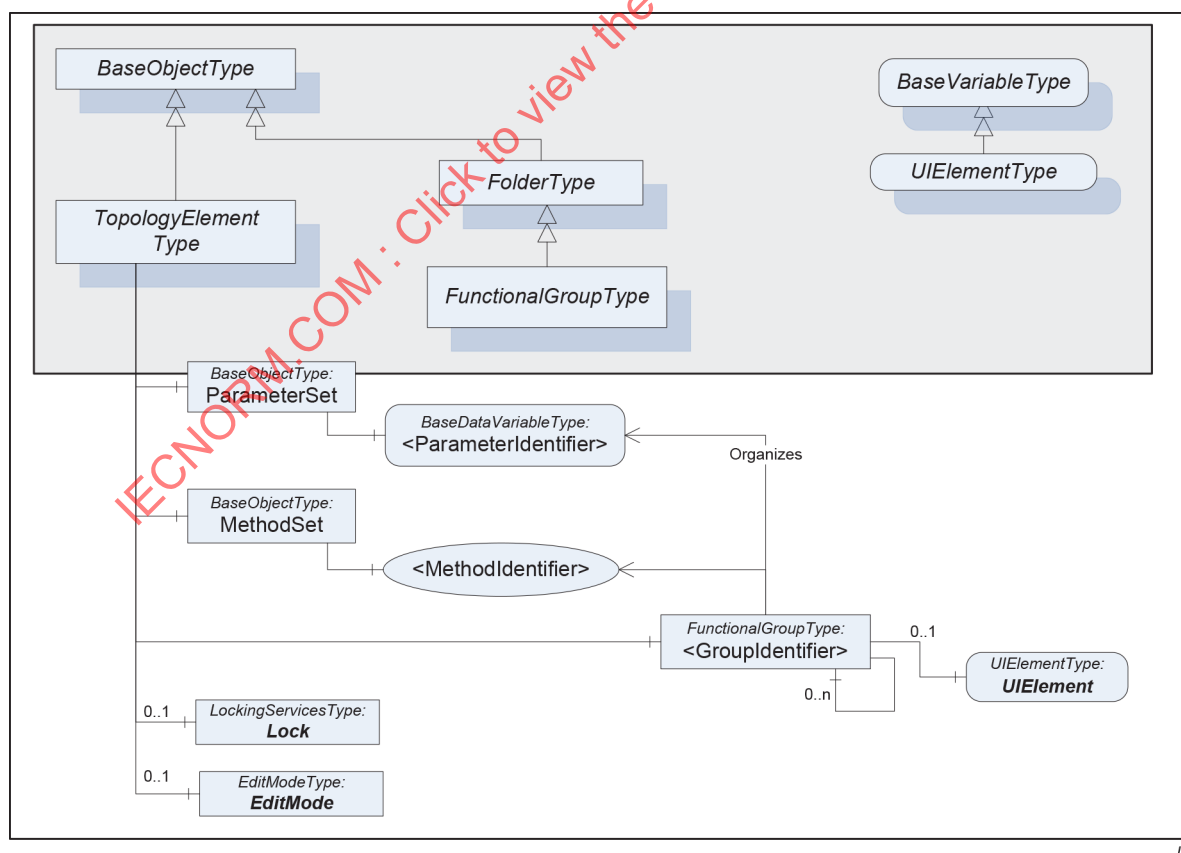
A.7.1.7 Accessibility and reliability

OPC UA defines a robust architecture with reliable communication mechanisms, configurable timeouts and automatic error detection.

A.7.2 OPC UA Information Models

The core of OPC UA is specified by IEC TR 62541-1 to IEC 62541-14. In particular, IEC 62541-5 defines the basic rules for the address space. The OPC UA address space is structured hierarchically, to foster the interoperability of clients and servers. The top levels are standardized for all servers. All nodes in the address space can be reached via the hierarchy. They can have additional references among each other, so that the address space forms a cohesive network of nodes. The OPC UA address space not only contains instances (instance space), but also the instance types (type space).

OPC UA for devices (IEC 62541-100) specifies in addition standardized structuring rules for components (see Figure A.11). Within components, parameters (IEC 61360 series properties) and methods can be hierarchically organized.



IEC

Figure A.11 – Component internal base mode

In addition, IEC 62541-100 defines modelling rules for composed components and network structures.

A.7.3 Relationship between AutomationML and OPC UA

By the means of a companion specification [6], a mapping between the contents expressed in AutomationML and OPC UA Information Models can be performed. This allows the dynamic exploration of contents which were defined in engineering life cycle phases, as well as, in a restricted manner, the maintenance and update of this model data ("as designed", "as built", "as operated").

IECNORM.COM : Click to view the full PDF of IEC 63278-1:2023

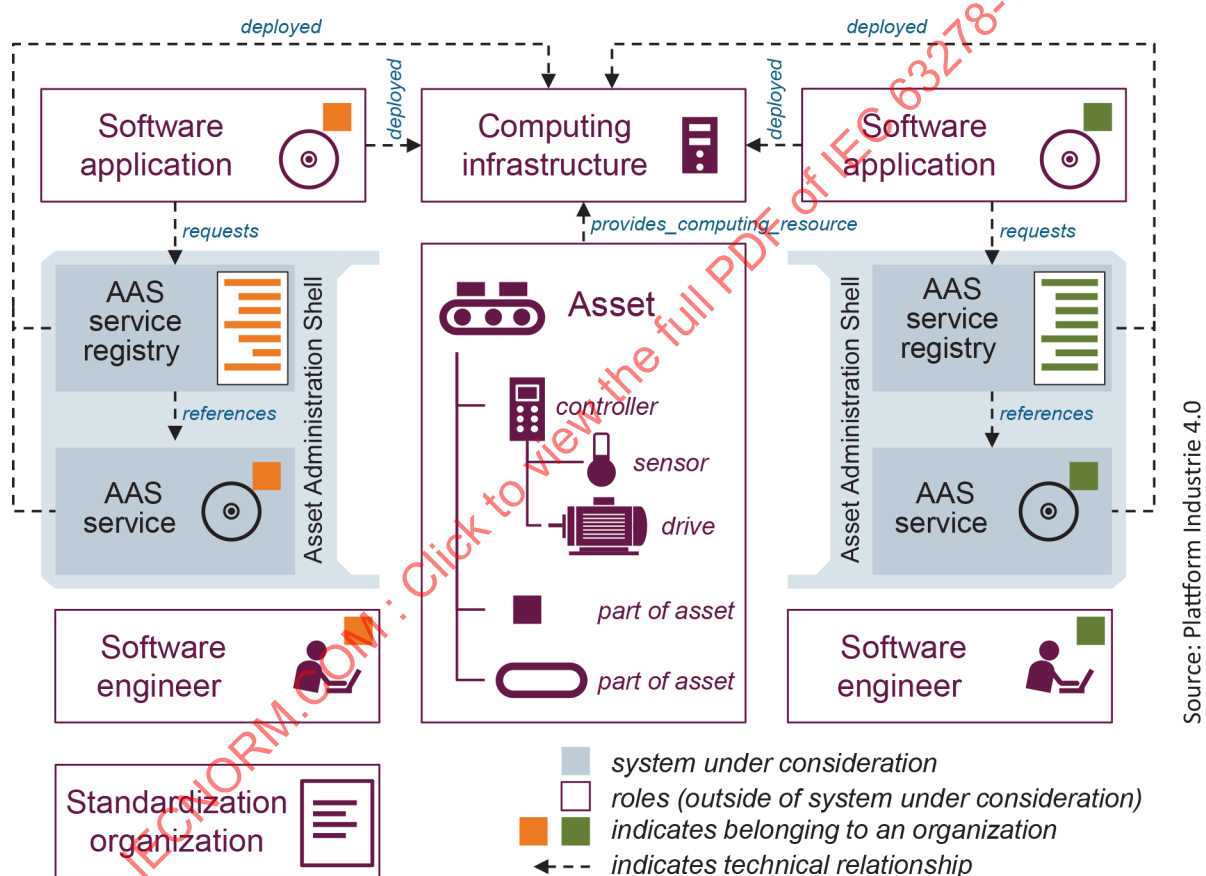
Annex B (informative)

Usage view of the Asset Administration Shell

Annex B aims at illustrating the relationship between the usage view of the Asset Administration Shell and the conceptual framework as defined in 4.1.

NOTE The usage view of the Asset Administration Shell is elaborated in [7].

Figure B.1 shows an overview of the usage view of the Asset Administration Shell. The system under consideration is shown in grey and the roles that are outside of the system boundary of the Asset Administration Shell are shown in dark purple. These roles interact with the Asset Administration Shell. The colour-filled rectangles indicate roles belonging to an organization which has an interest in the considered asset.



Source: Plattform Industrie 4.0

IEC

Figure B.1 – Overview of usage view of the Asset Administration Shell [7]

The Asset Administration Shell as a system under consideration conceptually includes the following constituents:

- **AAS service registry:** This role provides more specific aspects of the AAS interface in 4.2.7, especially with respect to the management of usage and access rights.
- **AAS service:** This role comprises specific implementation aspects of the Submodel in 4.2.8. Nevertheless, the principle of structuring Submodels by SubmodelElements in accordance with 4.2.9 is not considered in [7].

The following roles are considered:

- Asset: This role is identical to the asset in 4.2.3.
- Software engineer: This is a human or an organization implementing the interest in the asset by an AAS service registry, AAS services and software applications. Therefore, this role is more specific than the AAS responsible in 4.2.4.
- Software application: This role is identical to the AAS user application in 4.2.5.
- Computing infrastructure: The computing infrastructure provides capabilities to deploy and execute implementations of AAS service registries, AAS services, and software applications. The conceptual framework in Clause 4 does not address these aspects.
- Standardization organization: This is an organization with the objective to standardize certain AAS services and therefore can be compared to organizations publishing concept repositories (see A.1.2). The Submodel template as a result of a standardization activity is not considered as a separate role.

In summary, the usage view is a refinement of the conceptual framework according to 4.1.

IECNORM.COM : Click to view the full PDF of IEC 63278-1:2023

Annex C (informative)

Security for industrial automation and control systems

C.1 Security concepts from the IEC 62443 series

The IEC 62443 series provides a structured approach to derive security requirements for industrial automation and control systems (IACS).

The IEC 62443 series is based on the following concepts:

- Foundational requirements: The IEC 62443 series defines a total of seven Foundational Requirements with regards to the security of an IACS system (Clause C.2).
- Security levels: The IEC 62443 series defines security levels in terms of four different levels (SL1, SL2, SL3 and SL4), each with an increasing level of security (Clause C.3).

Clause C.2 and Clause C.3 provide further information on the concepts of the IEC 62443 series.

Whether the requirements are implemented by Asset Administration Shell built-in services or the Asset Administration Shell is used to enable and orchestrate the implementation of additional or complementary security measures is left open. In practical deployments both scenarios will exist, e.g., by considering a CNC program representing special knowledge being provided via an Asset Administration Shell. On the one hand the Asset Administration Shell can be used to specify the required protection level during processing by a trustworthy CNC drilling machine (e.g., the execution environment does provide SL3), on the other hand the integrity of this policy information itself needs to be properly protected within the Asset Administration Shell.

According to the IEC 62443 series the security requirements always depend on the risk associated with a threat identified for a specific usage scenario. Further explanation is given in C.4. The information model defined for the Asset Administration Shell security needs to provide the required types, structures and services to reflect these concepts. An instance of the Asset Administration Shell can use these structures depending on its needs.

C.2 Foundational requirements

The IEC 62443 series defines seven foundational requirements (FRs) that need to be addressed in order to implement cybersecurity for industrial automation and control systems. These foundational requirements also apply to smart manufacturing systems. The foundational requirements as given in IEC 62443-1-1:2009 are:

- FR1 Identification and Authentication Control (IAC): Identify and authenticate all users (humans, software processes and devices) before allowing them to access to the control system.
- FR2 Use Control (UC): Enforce the assigned privileges of an authenticated user (human, software process or device) to perform the requested action on the IACS and monitor the use of these privileges.
- FR3 System Integrity (SI): Ensure the integrity of the IACS to prevent unauthorized manipulation.
- FR4 Data Confidentiality (DC): Ensure the confidentiality of information on communication channels and in data repositories to prevent unauthorized disclosure.
- FR5 Restrict Data Flow (RDF): Segment the control system via zones and conduits to limit the unnecessary flow of data.

- FR6 Timely Response to Event (TRE): Respond to security violations by notifying the proper authority, reporting needed evidence of the violation and taking timely corrective action when incidents are discovered.
- FR7 Resource Availability (RA): Ensure the availability of the control system against the degradation or denial of essential services.

C.3 Security level

The security levels are defined as follows. Details can be found in IEC TS 62443-1-1:2009, 6.3.

- Security Level 1: Protection against casual or coincidental violation.
- Security Level 2: Protection against intentional violation using simple means with low resources, generic skills and low motivation.
- Security Level 3: Protection against intentional violation using sophisticated means with moderate resources, system specific skills and moderate motivation.
- Security Level 4: Protection against intentional violation using sophisticated means with extended resources, system specific skills and high motivation.

There are no definitions for Security Level 0 (meaning no security)

C.4 Measures of defence for IACS

The IEC 62443 series describes a process to define the security level in accordance with the foundational requirement for a given application using a risk-based approach.

Figure C.1 gives an example of the allocation of security levels to the individual foundational requirements applicable for a specific asset.

	SL 1	SL 2	SL 3	SL 4
FR 1 – Identification and Authentication Control	☐	☒	☐	☐
FR 2 – Use Control	☐	☒	☐	☐
FR 3 – System Integrity	☐	☐	☒	☐
FR 4 – Data Confidentiality	☒	☐	☐	☐
FR 5 – Restricted Data Flow	☐	☒	☐	☐
FR 6 – Timely Response to Events	☐	☒	☐	☐
FR 7 – Resource Availability	☐	☐	☒	☐

IEC

Figure C.1 – Foundational Requirements and Security Levels applicable for an Asset

Depending on the context different security levels can apply for individual assets analysed with respect to different foundational requirements.

Bibliography

Numbered references of documents in the text

- [1] Current Standards Landscape for Smart Manufacturing Systems, National Institute of Standards and Technology, U.S. Department of Commerce, 2016
- [2] Directive 2006/42/EC of the European Parliament and of the Council of 17 May 2006 on machinery, and amending Directive 95/16/EC (recast) (2006) Official Journal L157
- [3] Internet Engineering Task Force, 2005. RFC 3986: Uniform Resource Identifier (URI): Generic Syntax [online], final release January 2005, accessed: 27 February 2023, available from: <https://www.ietf.org/rfc/rfc3986.html>
- [4] DIN SPEC 91406:2019-12, Automatic identification of physical objects and information on physical objects in IT systems, particularly IoT systems
- [5] UNITED NATIONS / ECONOMIC COMMISSION FOR EUROPE; Recommendation No. 20; Codes for Units of Measure Used in International Trade – Revision 17 (Annexes I to III)
- [6] OPC 30040: OPC UA for AutomationML, Release 1.0, OPC Foundation, 22 Feb 2016
- [7] Usage view of the Administration Shell, Platform Industrie 4.0, February 2019

Standardisation documents referenced in the text

NOTE See also A.1.3

ISO/IEC Guide 77-2:2008, *Guide for specification of product properties and classes – Part 2: Technical principles and guidance*

ISO/IEC 6523:1998, *Information technology – Structure for the identification of organizations and organization parts – Part 1: Identification of organization identification schemes*

DIN 8580:2022-12, *Manufacturing processes – Terms and definitions, division*

ISO/IEC 11179-6:2015, *Information technology – Metadata registries (MDR) – Part 6: Registration*

ISO 18435-1:2009, *Industrial automation systems and integration – Diagnostics, capability assessment and maintenance applications integration – Part 1: Overview and general requirements*

ISO/IEC 20924:2021, *Internet of things (IoT) – Vocabulary*

ISO/IEC 21823-1:2019, *Internet of things (IoT) – Interoperability for IoT systems – Part 1: Framework*

ISO 22745 (all parts), *Industrial automation systems and integration – Open technical dictionaries and their application to master data*

ISO TR 23087:2018, *Automation systems and integration – The Big Picture of standards*

ISO/IEC 27000:2018, *Information technology – Security techniques – Information security management systems – Overview and vocabulary*

ISO 29002 (all parts), *Industrial automation systems and integration – Exchange of characteristic data*

IEC 61360-1:2017, *Standard data element types with associated classification scheme – Part 1: Definitions – Principles and methods*

IEC 61360-2:2012, *Standard data element types with associated classification scheme for electric components – Part 2: EXPRESS dictionary schema*

IEC 61406-1:2022, *Identification Link – Part 1: General requirements*

IEC 61666:2010+AMD1:2021 CSV, *Industrial systems, installations and equipment and industrial products – Identification of terminals within a system*

IEC 61987-10:2009, *Industrial-process measurement and control – Data structures and elements in process equipment catalogues – Part 10: List of Properties (LOPs) for Industrial-Process Measurement and Control for Electronic Data Exchange – Fundamentals*

IEC 61987-11:2016, *Industrial-process measurement and control – Data structures and elements in process equipment catalogues – Part 11: List of properties (LOPs) of measuring equipment for electronic data exchange – Generic structures*

IEC 61987-12:2016, *Industrial-process measurement and control – Data structures and elements in process equipment catalogues – Part 12: Lists of properties (LOPs) for flow measuring equipment for electronic data exchange*

IEC 61987-32, *Industrial-process measurement and control – Data structures and elements in process equipment catalogues – Part 32: Lists of properties (LOP) for I/O modules for electronic data exchange*

IEC 61987-92:2018, *Industrial-process measurement and control – Data structures and elements in process equipment catalogues – Part 92: Lists of properties (LOP) of measuring equipment for electronic data exchange – Aspect LOPs*

IEC TR 62390:2005, *Common automation device – Profile guideline*

IEC TS 62443-1-1:2009, *Industrial communication networks – Network and system security – Part 1-1: Terminology, concepts and models*

IEC 62541 (all parts), *OPC Unified Architecture*

IEC 62541-5:2020, *OPC Unified Architecture – Part 5: Information Model*

IEC 62541-100, *OPC Unified Architecture – Part 100: Device Interface*

IEC TR 62635:2012, *Guidelines for end-of-life information provided by manufacturers and recyclers and for recyclability rate calculation of electrical and electronic equipment*

IEC 62683-1, *Low-voltage switchgear and controlgear – Product data and properties for information exchange – Part 1: Catalogue data*

IEC 62832 (all parts), *Industrial-process measurement, control and automation – Digital factory framework*

IEC 62832-1:2020, *Industrial-process measurement, control and automation – Digital factory framework – Part 1: General principles*

IEC 62832-2:2020, *Industrial-process measurement, control and automation – Digital factory framework – Part 2: Model elements*

IEC 62890:2020, *Industrial-process measurement, control and automation – Life-cycle-management for systems and components*

IEC TR 63283-1:2022, *Industrial-process measurement, control and automation – Smart manufacturing – Part 1: Terms and definitions*

ISO/IEC TR 63306-1:2020, *Smart manufacturing standards map (SM2) – Part 1: Framework*

ISO/IEC TR 63306-2:2021, *Smart manufacturing standards map (SM2) – Part 2: Catalogue*

IECNORM.COM : Click to view the full PDF of IEC 63278-1:2023

SOMMAIRE

AVANT-PROPOS	78
INTRODUCTION.....	80
0.1 Généralités	80
0.2 Vue d'ensemble des parties de la série	80
0.3 Interopérabilité.....	81
0.4 Principaux objectifs de l'Enveloppe de Gestion d'Actif	82
1 Domaine d'application	83
2 Références normatives	83
3 Termes, définitions, abréviations et conventions	83
3.1 Termes et définitions	83
3.2 Termes abrégés.....	88
3.3 Conventions.....	90
4 Cadre conceptuel	90
4.1 Généralités	90
4.2 Enveloppe de Gestion d'Actif et entités connexes	91
4.2.1 Généralités	91
4.2.2 Vue d'ensemble détaillée.....	92
4.2.3 Actif.....	93
4.2.4 Responsable d'AAS	94
4.2.5 Application utilisateur d'AAS	94
4.2.6 Enveloppe de Gestion d'Actif.....	94
4.2.7 Interface d'AAS	95
4.2.8 Submodel	95
4.2.9 SubmodelElement.....	96
4.2.10 Modèle Submodel.....	96
4.2.11 Élément de modèle Submodel	97
4.2.12 Répertoires de concepts	97
4.2.13 Intégration d'actif.....	99
4.2.14 Service d'actif.....	100
4.2.15 Services relatifs à l'actif.....	100
4.3 Aspects du cycle de vie des actifs et des Enveloppes de Gestion d'Actif.....	100
4.4 Exemple d'un scénario global d'Enveloppe de Gestion d'Actif	103
5 Identificateurs.....	106
5.1 Besoins.....	106
5.2 Détermination des identificateurs	106
5.2.1 Généralités	106
5.2.2 Identificateurs distincts au niveau global pour les concepts par IRDI	107
5.2.3 Identificateurs distincts au niveau global par URI.....	107
5.2.4 Identificateurs locaux.....	107
6 Structure de l'Enveloppe de Gestion d'Actif	108
6.1 Généralités	108
6.2 Exigences associées à l'Enveloppe de Gestion d'Actif	109
6.2.1 Généralités	109
6.2.2 Enveloppe de Gestion d'Actif.....	109
6.2.3 Submodel	110
6.2.4 SubmodelElements.....	111

6.2.5	Interface d'AAS	115
6.3	Exigences relatives à l'exposition des données et à la sécurité de l'information	116
6.3.1	Exposition des données d'une Enveloppe de Gestion d'Actif et de ses Submodels	116
6.3.2	Exigences générales relatives à la sécurité de l'information de l'Enveloppe de Gestion d'Actif	117
6.3.3	Exigences relatives à la série IEC62443	118
Annexe A (informative) Normes correspondantes		119
A.1	Sources possibles pour les Submodels et les modèles Submodel	119
A.1.1	Généralités	119
A.1.2	Différents ensembles de définitions de concept pour les SubmodelElements	119
A.1.3	Spécifications internationales existantes pouvant être utilisées comme base pour les modèles Submodel	120
A.2	Dictionnaires, classes et types de propriétés de l'IEC 61360	124
A.2.1	Généralités	124
A.2.2	Classes	124
A.2.3	Types et instances de propriété	124
A.2.4	Dictionnaire de données communes de l'IEC (IEC CDD)	125
A.2.5	ECLASS	126
A.3	Classes et dictionnaire de la série IEC 61987	126
A.3.1	Généralités	126
A.3.2	Classes spécifiques	127
A.3.3	Dictionnaire	128
A.4	Classes et dictionnaire de la série IEC 62683	128
A.4.1	Généralités	128
A.4.2	Dictionnaire	129
A.5	Usine numérique (série IEC 62832)	129
A.5.1	Présentation de l'usine numérique	129
A.5.2	Compatibilité de l'usine numérique avec le concept d'Enveloppe de Gestion d'Actif	132
A.6	AutomationML (série IEC 62714)	134
A.6.1	Vue d'ensemble d'AutomationML	134
A.6.2	Concepts de modélisation AutomationML	136
A.6.3	Interopérabilité d'Enveloppe de Gestion d'Actif prise en charge par AutomationML	137
A.7	Architecture unifiée de l'OPC	138
A.7.1	Vue d'ensemble de l'architecture unifiée de l'OPC	138
A.7.2	Modèles d'information OPC UA	140
A.7.3	Relation entre AutomationML et l'architecture unifiée de l'OPC	141
Annexe B (informative) Vue d'exploitation de l'Enveloppe de Gestion d'Actif		142
Annexe C (informative) Sécurité pour les systèmes d'automatisation et de commande industriels		144
C.1	Concepts de sécurité issus de la série IEC62443	144
C.2	Exigences fondamentales	144
C.3	Niveau de sécurité	145
C.4	Mesures de défense pour l'IACS	146
Bibliographie		147

Figure 1 – Facettes d'interopérabilité conformément à l'ISO/IEC 21823-1	81
Figure 2 – Vue d'ensemble de l'Enveloppe de Gestion d'Actif et des entités connexes	91
Figure 3 – Échange d'informations entre applications utilisateur d'AAS.....	92
Figure 4 – Vue d'ensemble détaillée de l'Enveloppe de Gestion d'Actif et des entités connexes	92
Figure 5 – Actifs considérés comme actif de type ou actif d'instance	93
Figure 6 – Exemple d'Enveloppes de Gestion d'Actif différentes associées au même actif	95
Figure 7 – Exemple de différents types de répertoires de concepts référencés par SubmodelElements	98
Figure 8 – Concepts de premier niveau et relations de l'ontologie "Capability for Industry Ontology"	99
Figure 9 – Exemple de modélisation au moyen d'une Enveloppe de Gestion d'Actif	99
Figure 10 – Exemple d'intégration d'actif, de services d'actif et de services relatifs à l'actif.....	100
Figure 11 – Exemple d'aspects du cycle de vie des actifs et des Enveloppes de Gestion d'Actif	100
Figure 12 – Exemple d'actifs dans les cycles de vie des produits et des systèmes de production.....	101
Figure 13 – Exemple d'Enveloppe de Gestion d'Actif dans le cycle de vie du produit	102
Figure 14 – Représentation d'un exemple de scénario global d'Enveloppe de Gestion d'Actif	103
Figure 15 – Représentation de l'échange de valeurs dans le scénario global	105
Figure 16 – Différents identificateurs pour les identificateurs distincts au niveau global et les identificateurs locaux.....	107
Figure 17 – Exemple d'une Enveloppe de Gestion d'Actif démontrant la structure générale	108
Figure 18 – Représentation des différents aspects des SubmodelElements	112
Figure 19 – Enveloppes de Gestion d'Actif d'un ensemble représentatif d'axes électriques	114
Figure 20 – Application utilisateur d'AAS accédant aux interfaces d'AAS des Enveloppes de Gestion d'Actif	116
Figure 21 – Vue d'ensemble de la sécurité de l'Enveloppe de Gestion d'Actif	117
Figure A.1 – Exemple: Représentation d'un arbre de classe dans l'IEC CDD	124
Figure A.2 – Plan UML simplifié d'appareil, LOP et leurs aspects (IEC 61987-11).....	127
Figure A.3 – Organisation en blocs	129
Figure A.4 – Classe d'actif structurée dans l'usine numérique (voir la série IEC 62832).....	130
Figure A.5 – Description d'un actif structuré composé de plusieurs actifs.....	131
Figure A.6 – Comparaison des approches de l'usine numérique et de l'Enveloppe de Gestion d'Actif	132
Figure A.7 – Architecture d'AutomationML	135
Figure A.8 – Concepts CAEX	136
Figure A.9 – Relations entre éléments AutomationML	137
Figure A.10 – Organisation de la spécification OPC UA	139
Figure A.11 – Mode de base interne des composants	141
Figure B.1 – Vue d'ensemble de la vue d'exploitation de l'Enveloppe de Gestion d'Actif [7]	142

Figure C.1 – Exigences fondamentales et niveaux de sécurité applicables à un actif 146

Tableau 1 – Exemples de catégorisation des SubmodelElements 113

Tableau A.1 – Exemples de normes fournissant des entrées de répertoire de concepts qui peuvent être référencées par des modèles Submodel 121

Tableau A.2 – Exemples de normes fournissant des sources potentielles de modèles Submodel 122

Tableau A.3 – Exemples de normes fournissant des modèles de référence pour les Submodels 123

Tableau A.4 – Exemple de représentation d'un type de propriété avec certains attributs dans l'IEC CDD 125

Tableau A.5 – Comparaison des concepts individuels d'usine numérique (série IEC 62832) et d'Enveloppe de Gestion d'Actif 133

IECNORM.COM : Click to view the full PDF of IEC 63278-1:2023

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

ENVELOPPE DE GESTION D'ACTIF POUR APPLICATIONS INDUSTRIELLES –

Partie 1: Structure de l'Enveloppe de Gestion d'Actif

AVANT-PROPOS

- 1) La Commission Électrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. À cet effet, l'IEC — entre autres activités — publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments du présent document de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets.

L'IEC 63278-1 a été établie par le comité d'études 65 de l'IEC: Mesure, commande et automation dans les processus industriels. Il s'agit d'une Norme internationale.

Le texte de cette Norme internationale est issu des documents suivants:

Projet	Rapport de vote
65/1012/FDIS	65/1027/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à son approbation.

La langue employée pour l'élaboration de cette Norme internationale est l'anglais.

Ce document a été rédigé selon les Directives ISO/IEC, Partie 2, il a été développé selon les Directives ISO/IEC, Partie 1 et les Directives ISO/IEC, Supplément IEC, disponibles sous www.iec.ch/members_experts/refdocs. Les principaux types de documents développés par l'IEC sont décrits plus en détail sous www.iec.ch/publications.

Une liste de toutes les parties de la série IEC 63278, publiées sous le titre général *Enveloppe de Gestion d'Actif pour applications industrielles*, se trouve sur le site web de l'IEC.

Le comité a décidé que le contenu de ce document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous webstore.iec.ch dans les données relatives au document recherché. À cette date, le document sera

- reconduit,
- supprimé,
- remplacé par une édition révisée, ou
- amendé.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de ce document indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

IECNORM.COM : Click to view the full PDF of IEC 63278-1:2023

INTRODUCTION

0.1 Généralités

Le cycle de vie du système de production est axé sur la conception, le déploiement, la mise en service, le fonctionnement et le déclassement de la totalité d'une installation de production. La gestion du cycle de vie des produits est le processus de gestion de l'ensemble du cycle de vie d'un produit avec les flux d'information et les commandes depuis la création, en passant par le projet d'ingénierie et la fabrication, jusqu'à l'entretien et le traitement de fin de vie des produits manufacturés. La gestion de la chaîne d'approvisionnement est la gestion du flux de produits et services. Elle comprend les processus qui transforment les matières premières et les composants de pièces en produits finis. Elle implique la rationalisation des activités commerciales pour augmenter le plus possible la valeur du client et gagner un avantage concurrentiel sur le marché. Chacune de ces dimensions croise l'intégration verticale des machines, des usines et des systèmes d'entreprise dans la hiérarchie des équipements d'une pyramide d'entreprise. L'intégration d'applications logicielles industrielles dans chaque dimension et entre les dimensions permet d'effectuer des contrôles avancés dans l'atelier et de prendre des décisions optimales dans l'entreprise. Les détails des normes industrielles existantes pour chacune des trois dimensions du cycle de vie sont fournis dans la référence [1]¹.

Plusieurs technologies d'intégration ont été mises individuellement en pratique (par exemple CAO/FAO) visant à accélérer les cycles d'innovation de produits, à simplifier les chaînes d'approvisionnement et à augmenter la flexibilité des systèmes de production par l'échange d'informations entre les dimensions. Les détails des technologies d'intégration et des capacités prises en charge par celles-ci sont fournis dans la référence [1].

L'Enveloppe de Gestion d'Actif (AAS, de l'anglais *Asset Administration Shell*) est considérée comme une représentation interopérable d'un jumeau numérique dans la fabrication qui facilite une intégration plus poussée à l'intérieur et à travers les trois dimensions mentionnées ci-dessus.

Le présent document constitue la première partie de la série "Enveloppe de Gestion d'Actif pour applications industrielles". Les différentes parties de la série détaillent la structure, les modèles d'information, la définition des services et des interfaces en ligne, les aspects de sécurité exigés et les langages de communication, y compris le mapping du contenu des modèles d'architecture unifiée de l'OPC et AutomationML vers l'Enveloppe de Gestion d'Actif.

0.2 Vue d'ensemble des parties de la série

La planification actuelle prévoit des parties couvrant les sujets suivants:

- structure de l'Enveloppe de Gestion d'Actif (le présent document);
- métamodèle d'information (pour permettre d'accéder à des informations normalisées);
- dispositions de sécurité pour les Enveloppes de Gestion d'Actif;
- cas d'utilisation et exemples de modélisation
- interfaces vers les Enveloppes de Gestion d'Actif;
- langage de communication entre les ensembles d'Enveloppes de Gestion d'Actif;
- spécification du contenu des Enveloppes de Gestion d'Actif pour différents domaines.

¹ Les chiffres entre crochets renvoient à la Bibliographie.

La présente partie de l'IEC 63278 décrit les exigences relatives à la structure générale, auxquelles il convient que chaque Enveloppe de Gestion d'Actif possible satisfasse. Dans une partie suivante de la série, cette structure sera développée davantage vers un métamodèle de l'Enveloppe de Gestion d'Actif. Conformément à ces spécifications, des Enveloppes de Gestion d'Actif individuelles peuvent être créées. Ces Enveloppes de Gestion d'Actif individuelles constituent les conteneurs d'informations réels et fournissent des informations et des services concernant l'actif décrit.

0.3 Interopérabilité

L'objectif global de l'Enveloppe de Gestion d'Actif est la prise en charge de l'interopérabilité d'applications logicielles. Conformément à l'ISO/IEC 21823-1, différentes facettes d'interopérabilité peuvent être prises en compte (voir la Figure 1).

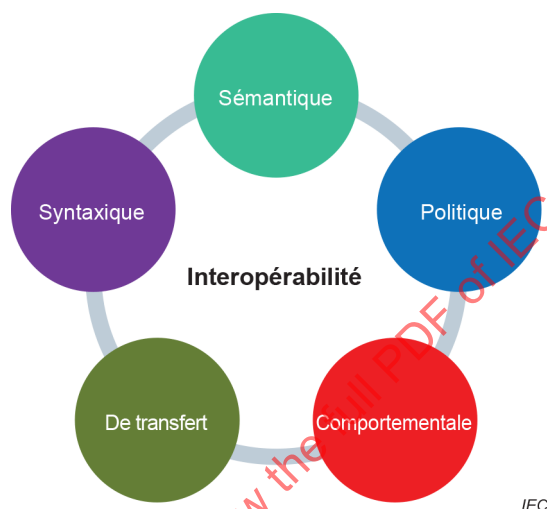


Figure 1 – Facettes d'interopérabilité conformes à l'ISO/IEC 21823-1

L'interopérabilité sémantique tient compte de la signification du modèle de données dans le contexte d'un sujet de sorte qu'il soit compris par les applications logicielles participantes. L'Enveloppe de Gestion d'Actif assure l'interopérabilité sémantique en associant des concepts bien connus aux données, qui sont échangées entre les applications logicielles.

L'interopérabilité politique concerne la conformité aux cadres juridiques, organisationnels et politiques applicables aux systèmes logiciels participants. L'Enveloppe de Gestion d'Actif assure l'interopérabilité politique de la manière suivante:

- L'Enveloppe de Gestion d'Actif assure une gestion uniforme de l'identité et du contrôle d'accès, y compris la restriction d'utilisation pour les informations et les services des actifs;
- L'Enveloppe de Gestion d'Actif permet une structuration uniforme des informations et des services des actifs. Cela permet à l'Enveloppe de Gestion d'Actif de définir et de maintenir la structure des informations et des services d'un actif et non des applications logicielles individuelles. Cela simplifie la gestion de l'information dans les industries de fabrication en réduisant l'effort et en augmentant la qualité de l'information.

L'interopérabilité de transfert considère le transfert de données entre applications logicielles sur la base d'une infrastructure de communication établie entre les applications logicielles participantes. Cette facette n'est pas abordée dans la présente partie de la série, mais elle sera prise en compte dans d'autres parties de la série.

L'interopérabilité syntaxique prend en compte le format de données par lequel les informations échangées peuvent être comprises par les applications logicielles participantes. Cette facette n'est pas abordée dans la présente partie de la série, mais elle sera prise en compte dans d'autres parties de la série.

L'interopérabilité comportementale prend en compte les résultats attendus des opérations d'interface. Cette facette est traitée par l'Enveloppe de Gestion d'Actif en ce sens que l'Enveloppe de Gestion d'Actif fournit une interface normalisée aux applications logicielles. Le comportement concret de cette interface normalisée sera pris en compte dans d'autres parties de la série.

0.4 Principaux objectifs de l'Enveloppe de Gestion d'Actif

Les déclarations suivantes résument ces discussions et formulent certains objectifs pour l'Enveloppe de Gestion d'Actif, contribuant ainsi à maintenir l'attention:

- L'Enveloppe de Gestion d'Actif vise à établir l'interopérabilité interentreprise.
Dans le cadre de la fabrication, les actifs sont fournis par de nombreuses entreprises différentes. Pour satisfaire aux scénarios d'aujourd'hui et de demain, il convient que les informations et les services relatifs aux actifs soient interopérables;
- L'Enveloppe de Gestion d'Actif est destinée aussi bien aux produits non intelligents qu'aux produits intelligents. Le concept d'actif comprend de nombreuses entités différentes, ayant ou n'ayant pas la capacité de communiquer activement ou d'être intelligentes. Pour tirer parti des avantages au niveau de l'ingénierie, de la maintenance ou de l'exploitation à tous les niveaux hiérarchiques, le concept de l'Enveloppe de Gestion d'Actif est approprié pour être appliqué par tous les actifs;
- L'Enveloppe de Gestion d'Actif vise à couvrir la totalité du cycle de vie des produits, appareils, machines et installations.
De nombreuses informations utiles sur les actifs sont constituées au début de leur cycle de vie, telles que la conception, l'ingénierie et la commercialisation. Pour maintenir l'efficacité économique, il convient de conserver et d'utiliser des informations numérisées issues de ces premières étapes dans les étapes ultérieures, telles que l'ingénierie des structures de niveau supérieur et l'exploitation et l'entretien de ces structures;
- L'Enveloppe de Gestion d'Actif vise à permettre des chaînes de valeur intégrées.
Les actifs pour les lignes de fabrication et les produits sont fournis par de nombreux partenaires différents de la chaîne de valeur. Pour maintenir l'efficacité économique, il convient d'échanger des informations numérisées entre les partenaires de la chaîne de valeur. Cela permet également des modes de production avancés (voir 0.1) ;
- L'Enveloppe de Gestion d'Actif est destinée à être une base pour les systèmes autonomes et l'intelligence artificielle.
À l'avenir, de nombreux avantages sont attendus des approches telles que les systèmes autonomes et l'intelligence artificielle. Ces approches exigent une base d'informations solide et des identificateurs d'éléments. L'Enveloppe de Gestion d'Actif fournit les deux.

ENVELOPPE DE GESTION D'ACTIF POUR APPLICATIONS INDUSTRIELLES –

Partie 1: Structure de l'Enveloppe de Gestion d'Actif

1 Domaine d'application

La présente partie de l'IEC 63278 définit la structure d'une représentation numérique normalisée d'un actif, appelée Enveloppe de Gestion d'Actif (AAS). L'Enveloppe de Gestion d'Actif donne un accès uniforme aux informations et aux services.

L'objectif de l'Enveloppe de Gestion d'Actif est de permettre à deux ou plusieurs applications logicielles d'échanger des informations et d'utiliser mutuellement les informations qui ont été échangées de manière sûre et fiable.

Le présent document est axé sur les Enveloppes de Gestion d'Actif représentant les actifs des entreprises de fabrication, y compris les produits fabriqués par ces entreprises et la hiérarchie complète des équipements industriels. Il définit les structures, les informations et les services connexes.

L'Enveloppe de Gestion d'Actif s'applique:

- à tout type de processus industriel (fabrication à l'unité, processus en continu, processus par lots, production hybride);
- à tout secteur industriel appliquant la mesure, la commande et l'automation dans les processus industriels;
- à la totalité du cycle de vie des actifs, de l'idée initiale au traitement de fin de vie;
- aux actifs qui sont des entités physiques, numériques ou immatérielles.

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 62443 (toutes les parties), *Sécurité des systèmes d'automatisation et de commande industriels*

3 Termes, définitions, abréviations et conventions

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions suivants s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse <https://www.electropedia.org/>
- ISO Online browsing platform: disponible à l'adresse <https://www.iso.org/obp>

3.1.1

actif

entité possédée par une organisation ou sous la garde de celle-ci, qui a une valeur perçue ou réelle pour cette organisation

Note 1 à l'article: Un actif peut être une entité unique, un ensemble d'entités un assemblage d'entités ou une composition d'entités.

EXEMPLE 1 Des exemples d'entités physiques sont les équipements, les matières premières, les composants de pièces et les pièces, les fournitures, les consommables, les produits physiques et les déchets.

EXEMPLE 2 Un logiciel est un exemple d'actif numérique.

EXEMPLE 3 Une licence logicielle est un exemple d'actif immatériel.

[SOURCE: IEC TR 63283-1:2022, 3.1.26, modifié — les notes et l'exemple ont été ajoutés]

3.1.2

Enveloppe de Gestion d'Actif

AAS

représentation numérique normalisée d'un actif

Note à l'article: L'abréviation "AAS" est dérivée du terme anglais développé correspondant « Asset Administration Shell ».

3.1.3

interface d'AAS

interface d'une AAS donnant un accès uniforme aux informations et aux services

EXEMPLE Des exemples de services sont les services d'exploration, les services d'actif et les services relatifs à l'actif.

3.1.4

responsable d'AAS

personne ou organisation étant intéressée par un actif et régissant une Enveloppe de Gestion d'Actif

3.1.5

application utilisateur d'AAS

application logicielle qui accède à une AAS par l'intermédiaire de son ou ses interfaces d'AAS, destinée à être utilisée par des humains ou à être soumise à un traitement automatique

3.1.6

intégration d'actif

logiciel ou infrastructure informatique, ou les deux, nécessaires pour accéder aux services d'actif

3.1.7

service d'actif

service fourni par l'actif considéré

3.1.8

service relatif à l'actif

service qui n'est pas fourni par l'actif considéré, mais par le logiciel ou l'infrastructure informatique, ou les deux, externes à l'actif considéré

3.1.9

composant

produit utilisé comme partie d'un produit assemblé, système ou installation industriel

[SOURCE: IEC 61666:2010+A1:2021 CSV, 3.6)

3.1.10**concept**

unité de connaissance créée par une combinaison unique de caractéristiques

[SOURCE: IEC 61360-1:2017, 3.1.8]

3.1.11**répertoire de concepts**

ensemble d'entrées permettant de faire une recherche par identificateur de concept et où la relation entre les entrées peut être décrite

[SOURCE: IEC 62832-1:2020, 3.1.5, modifié — le terme "dictionnaire de concepts" a été remplacé par "répertoire de concepts"; dans la définition, "dictionnaire de concepts" a été supprimé après "collection de"; la deuxième partie de la définition, commençant par "et où...", a été ajoutée et la note a été supprimée]

3.1.12**entrée du répertoire de concepts**

description d'un concept contenant au moins un identificateur de concept, un nom préférentiel et une description sans équivoque

[SOURCE: IEC 62832-1:2020, 3.1.6, modifié — dans le terme, "dictionnaire" a été remplacé par "répertoire" et dans la définition, "définition" a été remplacé par "description"]

3.1.13**représentation numérique**

informations et services représentant une entité d'un point de vue donné

EXEMPLE 1 Des exemples d'informations sont les propriétés (par exemple température maximale), les paramètres réels (par exemple vitesse réelle), les événements (par exemple notification de changement de statut), les schémas (électriques) et les informations de visualisation (plans 2D, plan 3D).

EXEMPLE 2 Les exemples de services sont des services d'actif (par exemple, fournir l'historique des données de configuration ou fournir la vitesse réelle) et des services relatifs à l'actif (par exemple, fournir une simulation).

EXEMPLE 3 Des exemples de points de vue sont des caractéristiques mécaniques, électriques ou commerciales.

3.1.14**traitement de fin de vie**

toute opération effectuée après qu'un déchet a été transféré à un établissement assurant la réutilisation de produits et de parties de produits, le recyclage des matériaux, la récupération d'énergie et l'élimination des résidus

Note 1 à l'article: Il comprend le démantèlement, le tri et l'élimination des matériaux.

[SOURCE: IEC/TR 62635:2012, 3.3, modifié — la Note 1 à l'article a été ajoutée]

3.1.15**entité**

chose (physique ou abstraite) ayant une existence spécifique

[SOURCE: ISO/IEC 20924:2021, 3.1.18]

3.1.16

secteur industriel

groupement basé sur des processus de production similaires, des produits similaires, des activités similaires ou un comportement similaire sur les marchés financiers

EXEMPLE Les exemples sont les suivants. Technologies de la santé, Environnement — Protection de la santé — Sécurité, Métrologie et mesure — Phénomènes physiques, Essais, Systèmes mécaniques et composants à usage général, Systèmes de fluides et composants pour usage général, Ingénierie de fabrication, Ingénierie de l'énergie et du transfert thermique, Ingénierie électrique, Électronique, Télécommunications — Ingénierie audio et vidéo, Technologies de l'information, Technologies de l'image, Mécanique de précision — Joaillerie, Ingénierie des véhicules routiers, Ingénierie ferroviaire, Construction de navires et structures marines, Ingénierie aéronautique et spatiale, Équipements de manutention de matériaux, Emballage et distribution de biens, Technologies du textile et du cuir, Industrie de l'habillement, Agriculture, Technologie alimentaire, Technologie chimique, Mines et minéraux, Pétrole et technologies connexes, Métallurgie, Technologie du bois, Industries du verre et de la céramique, Industries du caoutchouc et du plastique, Technologie du papier, Industries des peintures et des couleurs, Matériaux de construction et bâtiment, Génie civil, Affaires militaires — Génie militaire — Armement, Équipement domestique et commercial — Loisirs — Sports.

3.1.17

interface

frontière partagée entre deux entités définie par des caractéristiques fonctionnelles, des caractéristiques de signal, ou d'autres caractéristiques appropriées

3.1.18

interopérabilité

capacité de deux entités ou plus à échanger des éléments conformément à un ensemble de règles et de mécanismes mis en œuvre par une interface dans chaque entité, afin d'exécuter leurs tâches respectives

EXEMPLE 1 Des exemples d'entités sont les appareils, le matériel, les machines, les personnes, les processus, les applications, les unités logicielles, les systèmes et les entreprises.

EXEMPLE 2 Des exemples d'éléments sont l'information, le matériau, l'énergie, le contrôle, les actifs et les idées.

3.1.19

actif d'instance

actif spécifique qui est identifiable de manière unique

EXEMPLE Des exemples d'actifs d'instance sont un matériau, un produit, une pièce, un appareil, une machine, un logiciel, un système de commande ou un système de production.

3.1.20

instance de propriété

informations constituées au moins de l'identificateur d'un type de propriété et d'une valeur de propriété

Note 1 à l'article: Le concept de type et d'instance s'applique aux propriétés. En cas d'omission, le terme propriété fait référence aux types de propriétés.

Note 2 à l'article: Les instances de propriété ont une valeur qui peut être fournie par le fabricant ou un autre partenaire dans la chaîne de valeur. Parfois, une instance de propriété existe sans valeur spécifique, par exemple en faisant une déclaration d'existence.

Note 3 à l'article: Une instance de propriété est appelée paire valeur-proprieté dans certaines normes ou élément de données dans d'autres normes.

3.1.21**type de propriété**

paramètre défini adapté à la description et la différenciation de produits

Note 1 à l'article: Le concept de type et d'instance s'applique aux propriétés. En cas d'omission, le terme propriété fait référence aux types de propriétés.

Note 2 à l'article: Les types de propriétés sont définis dans les dictionnaires (comme le Dictionnaire de données communes de l'IEC ou ECLASS), et ils n'ont pas de valeur. Le type de propriété est également appelé type d'élément de données dans certaines normes.

[SOURCE: Guide ISO/IEC 77-2:2008, 2.18, modifié — ajout de "type" au terme, les notes ont été supprimées et remplacées par de nouvelles notes]

3.1.22**rôle**

ensemble de caractéristiques qui distinguent l'aptitude d'une entité à présenter un ensemble de comportements exigés

[SOURCE: ISO 18435-1:2009, 3.22]

3.1.23**service**

partie spécifique de la fonctionnalité qui est fournie par une entité par l'intermédiaire d'interfaces

[SOURCE: IEC 60050-741:2020, 741-01-28]

3.1.24**application logicielle**

élément fonctionnel logiciel qui est spécifique à la solution d'un problème de mesure et de commande dans les processus industriels

Note 1 à l'article: Une application logicielle peut être distribuée entre les ressources et peut communiquer avec d'autres applications logicielles.

[SOURCE: IEC TR 62390:2005, 3.1.2; modifié — dans le terme, "programme d'application" a été remplacé par "application logicielle"]

3.1.25**Submodel**

représentation d'un aspect d'un actif

3.1.26**SubmodelElement**

élément d'un Submodel

3.1.27**modèle Submodel**

modèle pour la représentation d'un aspect d'un actif

3.1.28**élément de modèle Submodel**

élément d'un modèle Submodel

3.1.29 système

éléments en interaction, corrélés ou interdépendants formant un ensemble complexe

[SOURCE: IEC TS 62443-1-1:2009, 3.2.123]

3.1.30 actif de type

représentation (abstraite) d'un ensemble d'actifs d'instance avec des caractéristiques et des fonctionnalités communes

Note 1 à l'article: L'ensemble d'actifs d'instance peut exister ou ne pas exister.

EXEMPLE Des exemples d'actifs de type sont le type de matériau, un type de produit, un type de pièce, un type d'appareil, un type de machine, un type de logiciel, un type de système de commande, un type de système de production.

3.2 Termes abrégés

AAS	Asset Administration Shell (Enveloppe de Gestion d'Actif)
AES	Advanced Encryption Standard (norme de chiffrement avancé)
AML	AutomationML, Automation Markup Language (langage de balisage d'automatisation)
API	Application Programming Interface (interface de programmation d'application)
BIM	Building Information Modelling (modélisation des informations de la construction)
CAO	Conception assistée par ordinateur
CAEX	Computer Aided Engineering Exchange (échange de données techniques assisté par ordinateur)
FAO	Fabrication assistée par ordinateur
CDEL	Collection of Data Elements (ensemble d'éléments de données)
CDV	Committee Draft for Vote (projet de comité pour vote)
CE	Conformité européenne
CNC	Computer Numerical Control (commande numérique par ordinateur)
CRE	Concept Repository Entry (entrée du répertoire de concepts)
DC	Data Confidentiality (confidentialité des données)
DF	Digital Factory (usine numérique)
DI	Data Identifier (identificateur de données)
DIN	Deutsches Institut für Normung (institut allemand de normalisation)
DIN SPEC	Spécification publiée par l'Institut allemand de normalisation
ECCMA	Electronic Commerce Code Management Association (association pour la gestion des codes commerciaux électroniques)
EDD	Electronic Device Description (description électronique de produit)
EDDL	Electronic Device Description Language (langage de description électronique de produit)
EN	European Norm (norme européenne)
PGI	Progiciel de gestion intégré
ETIM	Format de partage et d'échange de données sur les produits basé sur l'identification taxonomique
UE	Union européenne
FB	Function Block (bloc fonctionnel)
FDI	Field Device Integration (intégration des appareils de terrain)

FDIS	Final Draft International Standard (projet final de Norme internationale)
FDT	Field Device Tool (outils des dispositifs de terrain)
FR	Foundational Requirement (exigence fondamentale)
GUID	Globally Unique IDentifier (identifiant global unique)
HTTP	HyperText Transfer Protocol (protocole de transfert hypertexte)
IAC	Identification and Authentication Control (contrôle d'identification et d'authentification)
IACS	Industrial Automation and Control Systems (systèmes d'automatisation et de commande industrielles)
IC	Interface Class (classe d'interface)
TIC	Technologies de l'information et de la communication
ID	Identificateur
IDTA	International Digital Twin Association (organisation internationale d'utilisateurs de jumeaux numériques)
IE	Internal Element (élément interne)
IEC	International Electrotechnical Commission (Commission électrotechnique internationale)
IEC CDD	IEC Common Data Dictionary (dictionnaire de données communes de l'IEC — série IEC 61360)
IETF	Internet Engineering Task Force (groupe de travail sur l'ingénierie Internet)
IEV	International Electrotechnical Vocabulary (Vocabulaire électrotechnique international)
IH	Instance Hierarchy (hiérarchie d'instance)
IRDI	International Registration Data Identifier (identificateur international de données d'enregistrement)
ISMS	Information Security Management System (système de management de la sécurité de l'information)
ISO	International Organization for Standardization (Organisation internationale de normalisation)
TI	Technologies de l'information
JEITA	Japan Electronics and Information Technology industries Association (association de commerce japonaise pour l'électronique et les technologies de l'information et de la communication)
LOP	List of Properties (liste des propriétés)
BT	Basse tension
MDR	MetaData Registry (registre des métadonnées)
MES	Manufacturing Execution System (système d'exécution de la fabrication)
MOM	Manufacturing Operation Management (gestion des opérations de fabrication)
MOM	Manufacturing Operations Management (gestion des opérations de fabrication)
NID	Namespace IDentifier (identifiant d'espace de nom)
NSS	Namespace Specific String (chaîne spécifique à l'espace de noms)
OLOP	Operating List of Properties (liste de propriétés fonctionnelles)
Architecture unifiée de l'OPC Architecture unifiée de l'OPC maintenue par la Fondation OPC	
TO	Technologie opérationnelle
OTD	Open Technical Dictionary (dictionnaire technique ouvert)
PAS	Publicly Available Specification (spécification accessible au public)

PLM	Product Life Cycle Management (gestion du cycle de vie du produit)
PS	Production System (système de production)
RA	Resource Availability (disponibilité des ressources)
RAI	Identificateur d'autorité d'enregistrement
RC	Role Class (classe de rôle)
RDF	Restrict Data Flow (limitation du flux de données)
RFC	Request for Comments (document RFC)
RFID	Radio-Frequency Identification (identification de fréquence radioélectrique)
SI	System Integrity (intégrité du système)
SL	Niveau de sécurité
SOA	Service-Oriented Architecture (architecture orientée service)
SSL	Secure Sockets Layer (couche de sockets sécurisés)
SUC	System Unit Class (classe d'unité de système)
TCP	Transmission Control Protocol (protocole de commande de transmission)
TR	Technical Report (rapport technique)
TRE	Timely Response to Event (réponse opportune à l'événement)
TS	Technical Specification (spécification technique)
UC	Use Control (contrôle d'utilisation)
UML	Unified Modeling Language (langage de modélisation unifié)
CEE-ONU	Commission économique pour l'Europe des Nations unies
URI	Uniform Resource Identifier (identificateur universel)
URL	Uniform Resource Locator (adresse universelle)
URN	Uniform Resource Name (nom universel)
UUID	Universally Unique Identifier (identificateur unique universel)
VI	Version Identifier (identificateur de version)
XML	Extensible Markup Language (langage de balisage extensible)

3.3 Conventions

Pour certains termes, tels qu'Enveloppe de Gestion d'Actif ou SubmodelElement, la casse "start case" ou la casse Pascal est utilisée, afin de les distinguer du texte anglais.

Les paragraphes de l'Article 6 intitulés "exigences" définissent les exigences et les recommandations.

4 Cadre conceptuel

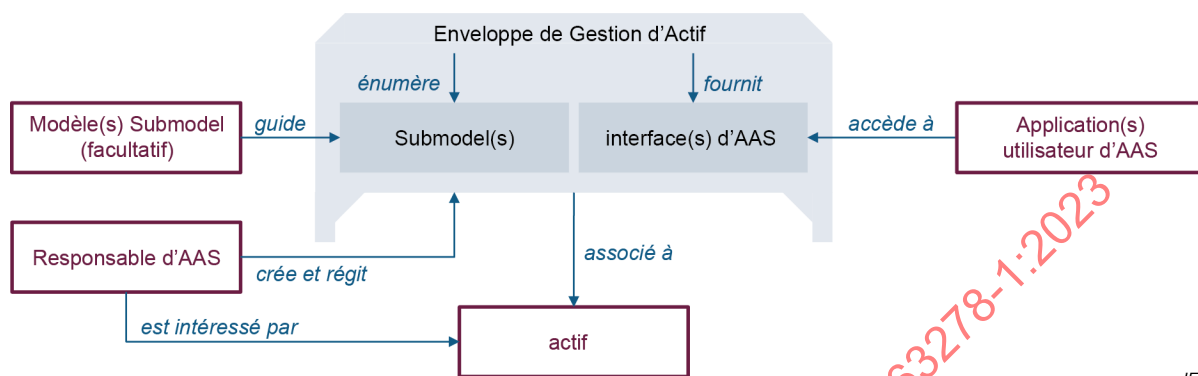
4.1 Généralités

Le cadre conceptuel décrit le contexte global de l'Enveloppe de Gestion d'Actif d'un point de vue de l'application. Cette description est nécessaire pour une compréhension globale de l'Enveloppe de Gestion d'Actif (voir également l'Annexe B). L'Article 4 est axé sur les permissions, les possibilités et les capacités, bien que peu d'exigences et de recommandations soient fournies, afin de décrire la perspective globale de l'application.

4.2 Enveloppe de Gestion d'Actif et entités connexes

4.2.1 Généralités

La Figure 2 représente une vue d'ensemble de l'Enveloppe de Gestion d'Actif et des entités connexes; ses parties principales sont en gris et les entités connexes en violet. La zone en grisé représente le domaine d'application de l'IEC 63278-1.



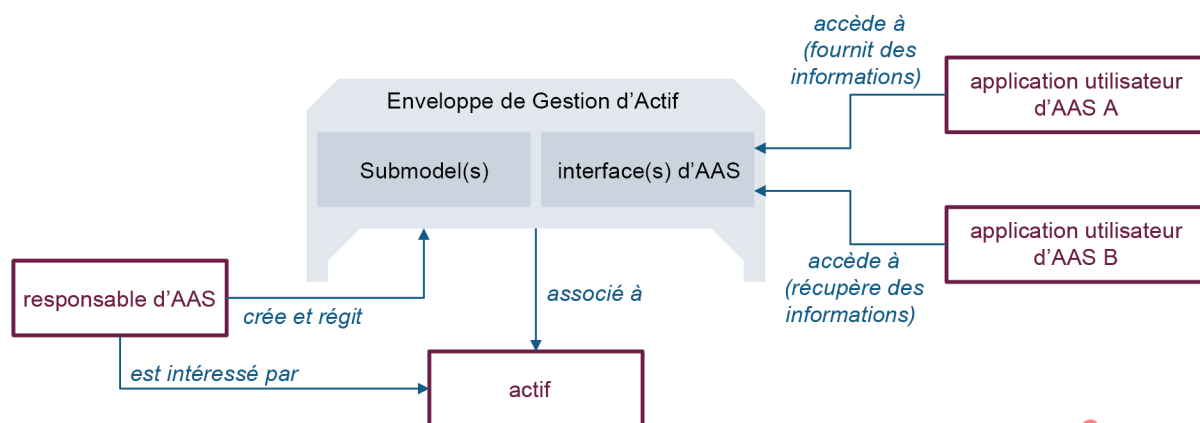
IEC

Figure 2 – Vue d'ensemble de l'Enveloppe de Gestion d'Actif et des entités connexes

Les relations suivantes entre les différentes entités représentées à la Figure 2 s'appliquent:

- une Enveloppe de Gestion d'Actif est associée à un actif;
- une Enveloppe de Gestion d'Actif fournit une ou plusieurs interfaces d'AAS;
- une Enveloppe de Gestion d'Actif énumère un ou plusieurs Submodels;
- un responsable d'AAS est intéressé par un actif et, en raison de cet intérêt, crée et régit une Enveloppe de Gestion d'Actif;
- une application utilisateur d'AAS accède à une Enveloppe de Gestion d'Actif par l'intermédiaire de son ou de ses interfaces d'AAS destinées à être utilisées par des humains ou pour un traitement automatisé;
- un modèle Submodel peut servir de recommandations pour un Submodel.

La Figure 2 représente également le fait qu'une application utilisateur d'AAS accède aux informations, qui sont représentées par des Submodels de l'Enveloppe de Gestion d'Actif, par l'intermédiaire de l'interface d'AAS de l'Enveloppe de Gestion d'Actif. La Figure 3 complète cela et représente la manière dont deux applications utilisateur d'AAS différentes peuvent échanger des informations relatives à un actif. Une application utilisateur d'AAS fournit les informations relatives à l'actif, par exemple la date de livraison, par l'intermédiaire de l'interface d'AAS dans l'Enveloppe de Gestion d'Actif, et une autre application utilisateur d'AAS récupère ces informations par l'intermédiaire de l'interface d'AAS. Le responsable d'AAS peut restreindre l'accès séparément pour chaque application utilisateur d'AAS.



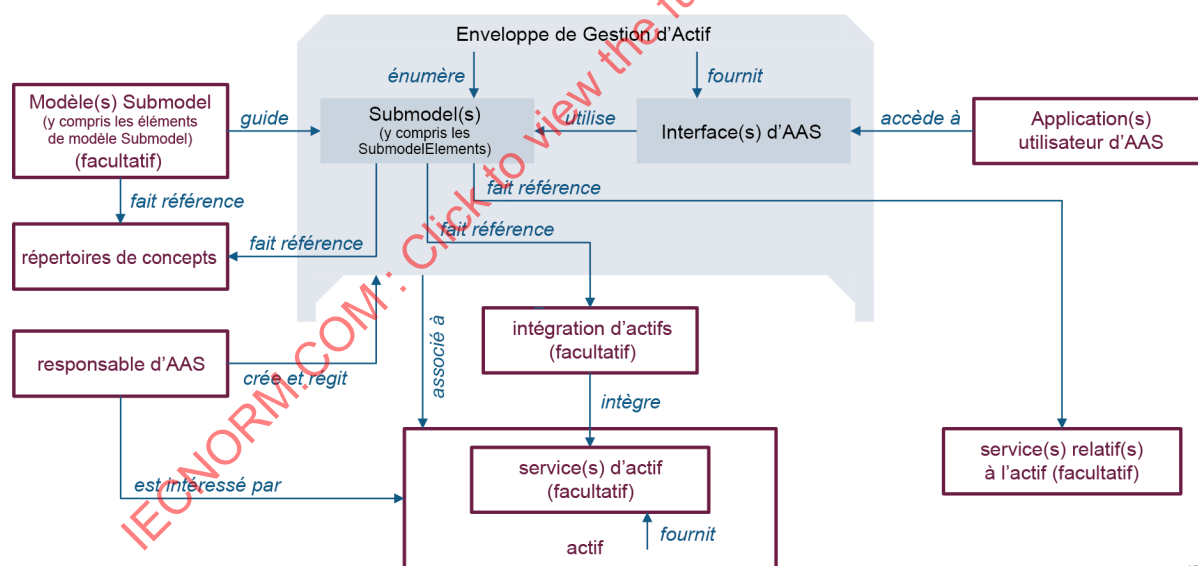
IEC

Figure 3 – Échange d'informations entre applications utilisateur d'AAS

De plus, des scénarios plus complexes sont possibles. Par exemple, différents responsables d'AAS peuvent définir leurs propres Enveloppes de Gestion d'Actif différentes pour un actif (voir la Figure 6) et définir l'accès par les applications utilisateur d'AAS de leur point de vue. Ainsi, différents cas d'utilisation peuvent être mis en œuvre pour un échange d'informations interentreprise.

4.2.2 Vue d'ensemble détaillée

La Figure 4 représente une vue d'ensemble plus détaillée de l'Enveloppe de Gestion d'Actif.



IEC

NOTE Les flèches de cette figure expriment uniquement le fait que des relations existent. Les autres parties de la série IEC 63278 peuvent être plus spécifiques.

Figure 4 – Vue d'ensemble détaillée de l'Enveloppe de Gestion d'Actif et des entités connexes

En plus des relations représentées à la Figure 2, la Figure 4 représente les relations supplémentaires suivantes:

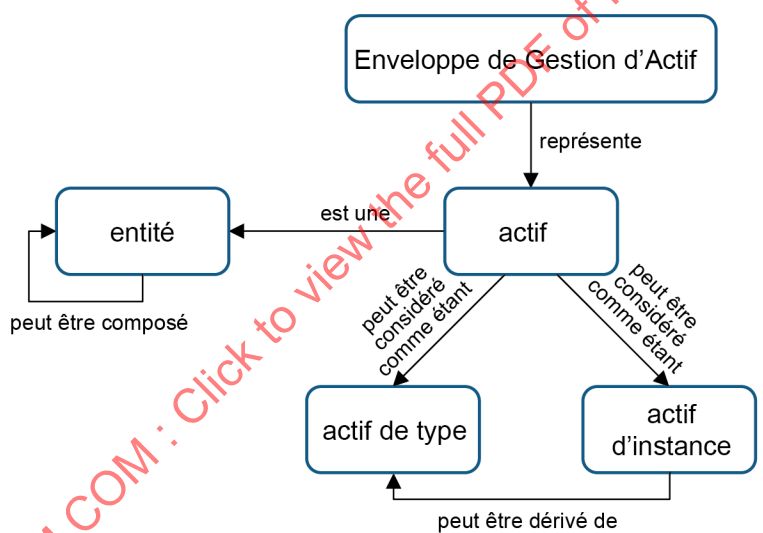
- répertoires de concepts de référence de Submodels;
- un modèle Submodel peut être utilisé pour servir de recommandations pour un Submodel et des répertoires de concepts de référence;
- un actif peut fournir des services d'actif;
- les Submodels peuvent faire référence aux services d'actif par l'intermédiaire d'une intégration d'actif et peuvent également faire référence à des services relatifs à l'actif.

Tous les éléments représentés à la Figure 4 sont expliqués dans les paragraphes 4.2.3 à 4.2.15.

4.2.3 Actif

Un actif est utile pour une personne ou une organisation, c'est pourquoi il est représenté par une Enveloppe de Gestion d'Actif. Seul le responsable d'AAS détermine la nature d'un actif; le présent document n'établit aucune spécification concernant la nature d'un actif.

Un actif possède un identificateur unique.



IEC

Figure 5 – Actifs considérés comme actif de type ou actif d'instance

La Figure 5 représente différents types d'actifs et de relations entre différents actifs. Par exemple:

- un actif peut être une entité simple, par exemple un composant unique, qui n'est pas composé d'autres entités;
- un actif peut être composé de différentes entités. Ces entités peuvent être des actifs qui sont représentés par une Enveloppe de Gestion d'Actif ou des entités qui ne sont pas représentées par une Enveloppe de Gestion d'Actif. Par exemple, une machine peut contenir différents entraînements qui peuvent être représentés par des Enveloppes de Gestion d'Actif;

- il peut exister une relation "dérivé de" entre un actif de type et un actif d'instance, puisqu'un actif peut être considéré comme un actif de type ou comme un actif d'instance.

EXEMPLE Pour de nombreux actifs, la distinction entre actif de type et actif d'instance peut être appliquée. Les fabricants de produits, d'équipements d'automatisation et de systèmes de production utilisent souvent des actifs de type pour définir un ensemble commun de caractéristiques proposées sur le marché. Un actif de type peut être utilisé pour dériver des actifs d'instance spécifiques avec des caractéristiques clairement définies. Par exemple, un constructeur de machines réalise une machine particulière (actif d'instance) selon un type de machine (actif de type).

Souvent, les actifs d'instances multiples sont dérivés du même actif de type. Le responsable d'AAS définit si un actif est considéré comme un actif de type ou un actif d'instance.

Il existe également des relations technologiques entre actifs, non représentées à la Figure 5, par exemple une pompe qui alimente un réservoir, un entraînement qui est raccordé électriquement à une alimentation, ou deux pièces qui sont vissées l'une à l'autre.

L'Enveloppe de Gestion d'Actif est par conséquent en mesure de représenter ces différents types de relations (voir 4.2.9).

NOTE L'actif défini dans le présent document est un super-ensemble de l'actif PS défini dans la série IEC 62832. Dans le présent document, l'actif inclut également des entités qui ne font pas partie du système de production.

4.2.4 Responsable d'AAS

Un responsable d'AAS définit le domaine d'application de l'Enveloppe de Gestion d'Actif et crée et régit l'AAS. En général, un responsable d'AAS ne définit qu'une seule Enveloppe de Gestion d'Actif pour un actif, en se concentrant sur les actifs comme points de référence principaux pour les informations de structuration.

Un responsable d'AAS possède un identificateur unique.

Le responsable d'AAS attribue des droits d'accès à une application utilisateur d'AAS, c'est-à-dire qu'une application utilisateur d'AAS peut ne pas être en mesure d'accéder à toutes les informations ou peut ne pas être autorisée à exécuter tous les services proposés par l'intermédiaire de la ou des interfaces d'AAS.

EXEMPLE Les exemples de responsables d'AAS sont le constructeur de la machine ou un utilisateur de la machine (voir la Figure 6).

4.2.5 Application utilisateur d'AAS

L'objectif de l'Enveloppe de Gestion d'Actif est de permettre à deux ou plusieurs applications logicielles d'échanger des informations. Ces applications logicielles sont des applications utilisateur d'AAS représentées à la Figure 4.

Une application utilisateur d'AAS possède un identificateur unique.

Le développement ou l'exécution d'une application utilisateur d'AAS peut être guidé par un modèle Submodel.

4.2.6 Enveloppe de Gestion d'Actif

L'Enveloppe de Gestion d'Actif donne un accès uniforme aux informations et aux services de l'actif ou relatifs à l'actif, ce qui facilite l'interopérabilité entre les applications utilisateur d'AAS.

Une Enveloppe de Gestion d'Actif fournit une ou plusieurs interfaces d'AAS.

Une Enveloppe de Gestion d'Actif possède un identificateur unique.

Il existe des Submodels énumérés dans une Enveloppe de Gestion d'Actif.

Une Enveloppe de Gestion d'Actif est créée et régie par un responsable d'AAS et est reliée à un actif.

Les relations entre différentes Enveloppes de Gestion d'Actif peuvent être définies:

- s'il existe une relation entre différents actifs et si ces actifs sont représentés par des Enveloppes de Gestion d'Actif, la relation entre les actifs peut être représentée par une relation entre les Enveloppes de Gestion d'Actif correspondantes;
- les relations entre les Enveloppes de Gestion d'Actif peuvent être définies, même s'il n'y a pas de relation entre les actifs associés;
- une relation "dérivé de" peut être définie entre les Enveloppes de Gestion d'Actif d'un actif de type et d'un actif d'instance.

Ces relations sont définies par des SubmodelElements avec l'objectif de décrire des relations (voir 4.2.9).

Une Enveloppe de Gestion d'Actif doit être associée à un actif, qui peut être considéré comme un actif de type ou comme un actif d'instance.

Plusieurs responsables d'AAS peuvent être intéressés par le même actif, par exemple le fabricant d'une machine et l'utilisateur d'une machine (voir la Figure 6). Il peut en résulter plusieurs Enveloppes de Gestion d'Actif avec leurs responsables d'AAS correspondants associés au même actif. Cependant, cela signifie également que différents responsables d'AAS peuvent attribuer différents ensembles de caractéristiques à un seul actif.

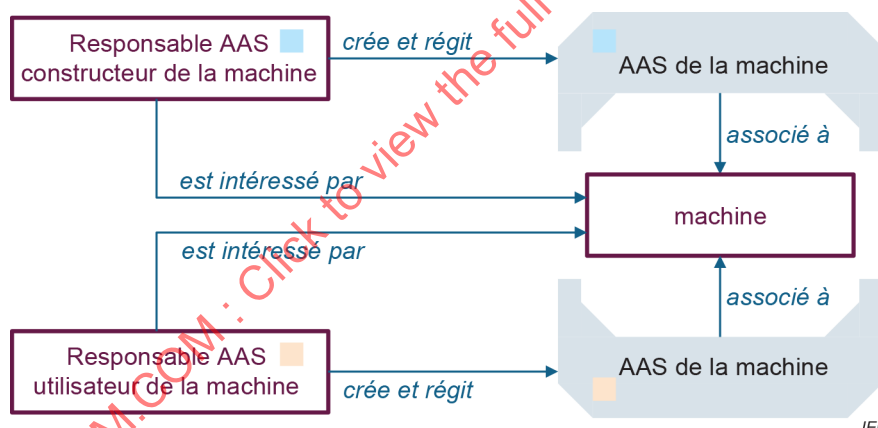


Figure 6 – Exemple d'Enveloppes de Gestion d'Actif différentes associées au même actif

4.2.7 Interface d'AAS

L'objectif de l'interface d'AAS est de permettre à une application utilisateur d'AAS d'accéder aux informations et aux services de l'actif ou liés à l'actif.

Une Enveloppe de Gestion d'Actif peut avoir plusieurs interfaces d'AAS.

Il est possible de permettre à plusieurs applications utilisateur d'AAS d'accéder aux informations d'une même Enveloppe de Gestion d'Actif par l'intermédiaire de son ou de ses interfaces d'AAS. L'ensemble d'informations auquel il est possible d'accéder peut varier pour différentes applications utilisateur d'AAS (voir 4.2.4).

4.2.8 Submodel

Un Submodel énuméré dans une Enveloppe de Gestion d'Actif représente les informations et décrit les services et représente ainsi un aspect spécifique d'un actif.

L'Enveloppe de Gestion d'Actif permet l'exécution des services décrits par un Submodel par l'intermédiaire de l'interface d'AAS et l'AAS accède aux services relatifs à l'actif et utilise l'intégration d'actif pour accéder aux services d'actif.

Un Submodel possède un identificateur unique.

Un Submodel peut utiliser un modèle Submodel comme modèle.

EXEMPLE Un exemple de Submodel est la plaque signalétique numérique. La directive 2006/42/CE [2] de l'Union européenne vise à normaliser les exigences d'accès au marché pour les machines dans l'Union européenne et d'autres pays associés. En ce qui concerne la plaque signalétique, la directive UE établit les exigences minimales relatives aux informations qu'une plaque signalétique doit inclure:

- la raison sociale et l'adresse complète du fabricant et, le cas échéant, de son mandataire;
- la désignation de la machine;
- le marquage CE;
- la désignation de la série ou du type;
- le numéro de série, le cas échéant;
- l'année de construction, c'est-à-dire l'année au cours de laquelle le processus de fabrication a été achevé.

La plaque signalétique numérique combine toutes ces informations dans un Submodel, voir <https://www.zvei.org/presse-medien/publikationen/submodel-templates-of-the-asset-administration-shell-zvei-digital-nameplate-for-industrial-equipment-version-10>.

NOTE Un Submodel tel que défini par le présent document équivaut à un type spécial d'ensemble d'éléments de données (CDEL), tel que défini dans la série IEC 62832. Le présent document définit des règles supplémentaires pour un Submodel.

4.2.9 SubmodelElement

Avec un SubmodelElement, des objectifs différents peuvent être poursuivis, par exemple:

- la représentation des informations relatives à un actif;
- la description des relations entre entités;
- la description des services et l'activation de l'exécution des services, par exemple les services d'actif ou les services relatifs à l'actif. La manière dont ces services sont exécutés est considérée comme un aspect de la mise en œuvre et ne relève pas du domaine d'application du présent document;
- la structuration hiérarchique des Submodels.

NOTE Les relations sont considérées comme un type particulier d'informations.

Un SubmodelElement peut faire référence à un élément de modèle Submodel en tant que modèle.

Si un SubmodelElement ne fait pas référence à un élément de modèle Submodel, une référence à un concept associé est nécessaire.

4.2.10 Modèle Submodel

Un modèle Submodel spécifie la structure d'un Submodel.

Un modèle Submodel possède un identificateur unique.

Les modèles Submodel guident la création de Submodels. Pour être acceptés publiquement, ces modèles sont souvent définis conformément aux Normes internationales existantes (voir A.1.3).

4.2.11 Élément de modèle Submodel

Un élément de modèle Submodel spécifie la structure d'un SubmodelElement.

Un élément de modèle Submodel spécifie la structure en fonction de son objectif. L'objectif est défini par le concept référencé, par exemple des concepts décrivant:

- des informations relatives à un actif;
- une relation;
- un service;
- une structure.

Un élément de modèle Submodel spécifie également la cardinalité dans laquelle un SubmodelElement doit être présent.

4.2.12 Répertoires de concepts

Les référentiels de concepts servent à définir des vocabulaires communs et à décrire les relations au sein de ces vocabulaires. Un vocabulaire commun est la base pour permettre à deux ou plusieurs applications logicielles d'échanger des informations et d'utiliser mutuellement les informations échangées.

Ces répertoires de concepts peuvent être définis par des normes, des spécifications de consortium, des spécifications du fabricant et des spécifications de communauté ouverte (voir la Figure 7 et A.1.2). Les entrées de répertoire de concepts peuvent être référencées par les modèles Submodel et les Submodels d'une Enveloppe de Gestion d'Actif. S'il y a des modifications dans un répertoire de concepts, il incombe alors au propriétaire du modèle Submodel et au responsable d'AAS de répondre de manière appropriée. Le responsable d'AAS, le propriétaire d'un modèle Submodel et le fournisseur d'un répertoire de concepts peuvent être des organisations différentes. Par conséquent, les répertoires de concepts, les modèles Submodel et les Submodels sont gérés de manière indépendante.

EXEMPLE 1 Le Common Data Dictionary (CDD) (dictionnaire de données communes) de l'IEC est un exemple de répertoire de concepts détenu et géré par l'IEC. La plaque signalétique numérique est un exemple pour un modèle Submodel géré par l'IDTA, tandis qu'une société de fabrication utilise le modèle Submodel comme recommandations pour les Submodels de leurs Enveloppes de Gestion d'Actif.

NOTE Voir l'IEC 62832-2:2020, Figure D.6 pour les similitudes.

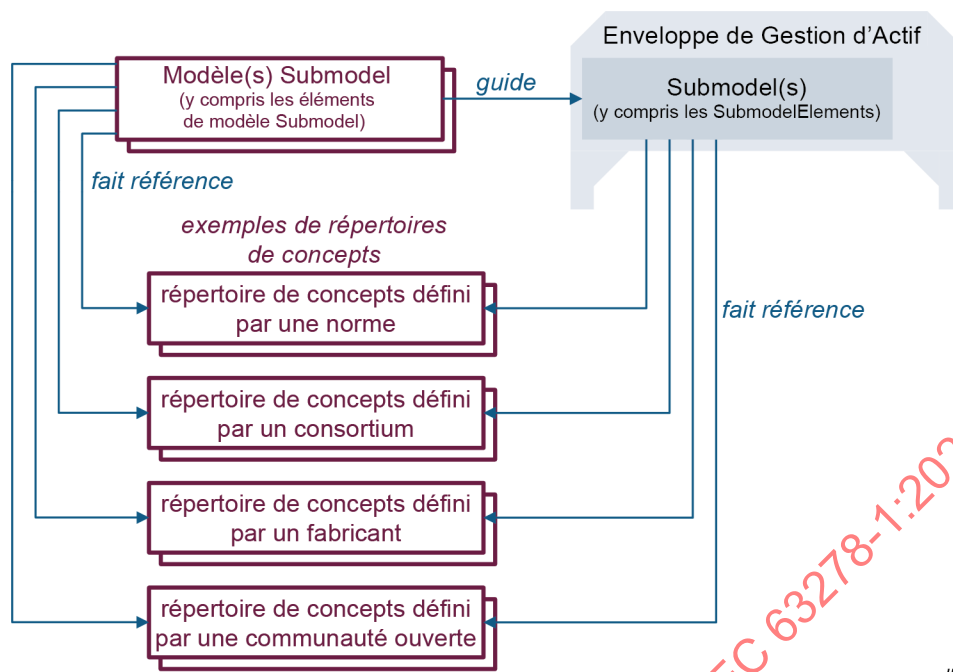


Figure 7 – Exemple de différents types de répertoires de concepts référencés par SubmodelElements

EXEMPLE 2 Les dictionnaires de concepts tels que l'ECLASS ou l'IEC CDD et les ontologies telles que les ontologies de référence ouvertes de l'Industrial Ontologies Foundry sont des exemples de répertoires de concepts, voir <https://www.industrialontologies.org/>, ou de l'Open Biological and Biomedical Ontology, voir <https://obofoundry.org/>.

Les répertoires de concepts peuvent être consultés par le responsable d'AAS pendant la phase de conception d'un Submodel d'une Enveloppe de Gestion d'Actif et être accessibles par une application utilisateur d'AAS pendant la phase d'utilisation d'une Enveloppe de Gestion d'Actif.

- Au cours de la phase de conception, les répertoires de concepts existants peuvent être entrés pour donner au responsable d'AAS une base pour la conception d'un Submodel d'une Enveloppe de Gestion d'Actif et lui permettre de décider de la manière d'utiliser et d'intégrer un répertoire de concepts.
- Dans la phase d'utilisation, une application utilisateur d'AAS accède aux informations et aux services, qui sont représentés par un Submodel. Une application utilisateur d'AAS peut également accéder aux définitions de concept fournies par le répertoire de concepts s'ils sont référencés par des SubmodelElements d'un Submodel.

L'utilisation des répertoires de concepts permet aux informations et aux services fournis par une Enveloppe de Gestion d'Actif d'être basés sur des définitions de concept normalisées.

EXEMPLE 3 Une application possible des Enveloppes de Gestion d'Actif est la fourniture d'informations aux applications utilisateur d'AAS pour rationaliser la recherche de correspondances entre les capacités exigées et les compétences disponibles. Les produits à produire exigent des capacités, qui sont décrites comme étant neutres en ressources. Les ressources industrielles sont représentées par des Enveloppes de Gestion d'Actif offrant des capacités neutres en ressources, des compétences disponibles et des relations entre capacités et compétences. L'ontologie "Capability for Industry" décrit ces concepts de premier niveau et ces relations (voir la Figure 8). Une application utilisateur d'AAS utilise ces informations pour raisonner, en se basant sur une connaissance sémantique particulière des processus de fabrication, par exemple, qui sont spécifiés par une ontologie spécifique à un domaine tel que DIN 8580.

La Figure 9 donne un exemple d'Enveloppe de Gestion d'Actif d'une perceuse. Dans un Submodel, un SubmodelElement pour la capacité trou profond est donné, qui fait référence à une CRE trou profond, spécifié par exemple par ECLASS. Dans un autre Submodel de la même Enveloppe de Gestion d'Actif, une compétence perçage est décrite et est liée à la capacité par un SubmodelElement dédié avec une relation d'objectif. Pour que la commande soit précise, une certaine variable pour la profondeur de perçage est décrite, qui fait référence à une entrée de répertoire de concepts également spécifiée par ECLASS. Le raisonnement exécuté par une application utilisateur d'AAS est présumé définir cette profondeur de perçage en fonction des caractéristiques appliquées à la capacité neutre en ressources.

Cet exemple représente également la manière dont des schémas existants, tels que les spécifications OPC UA, peuvent être intégrés à l'Enveloppe de Gestion d'Actif.

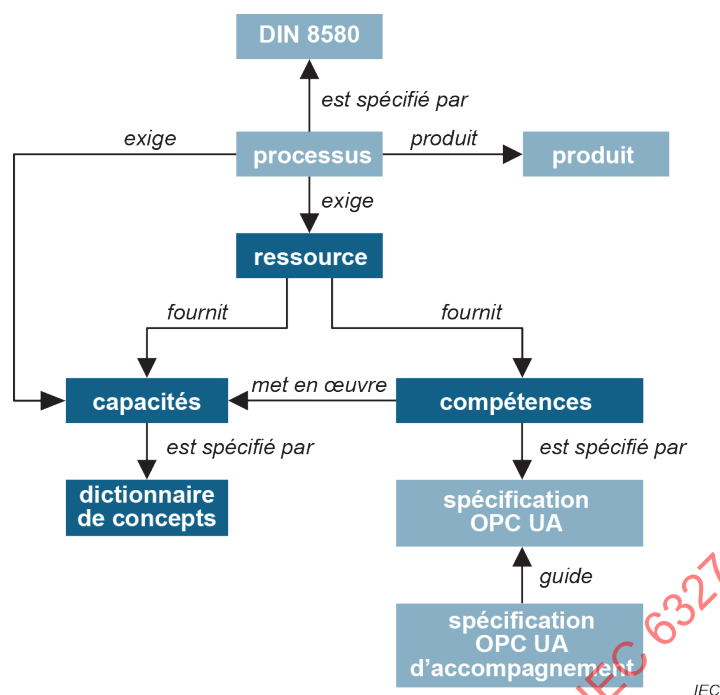


Figure 8 – Concepts de premier niveau et relations de l'ontologie "Capability for Industry Ontology"

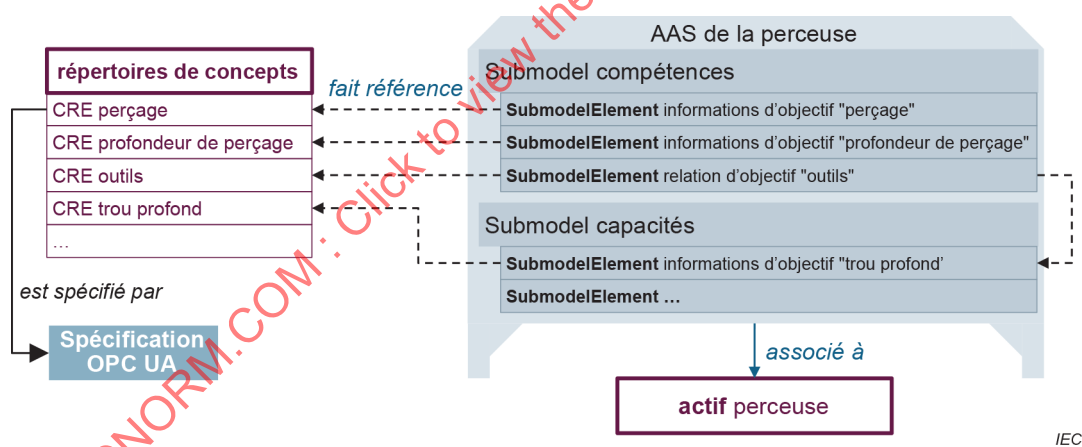


Figure 9 – Exemple de modélisation au moyen d'une Enveloppe de Gestion d'Actif

4.2.13 Intégration d'actif

Une intégration d'actif sert de moyen technique d'intégration et d'accès, de sorte que tout service d'actif puisse être uniformément accessible à partir d'une Enveloppe de Gestion d'Actif.

EXEMPLE Un appareil périphérique supplémentaire est intégré à une machine fournissant des services d'actif sélectionnés de la machine. L'appareil périphérique met en œuvre une intégration d'actif pour mettre ces services d'actif à la disposition des Submodels associés (voir la Figure 10). Des exemples de services d'actif sont de fournir la valeur réelle d'une machine et de configurer le logiciel d'une machine. Des exemples de services relatifs à l'actif sont de fournir la documentation d'une machine ou d'un modèle de simulation d'une machine, si les services ne sont pas fournis par l'actif lui-même, par exemple dans le cas d'un composant purement mécanique.

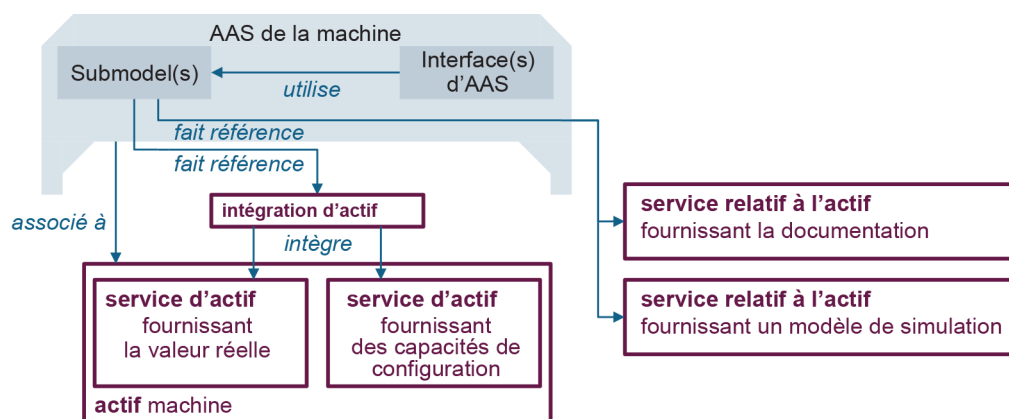


Figure 10 – Exemple d'intégration d'actif, de services d'actif et de services relatifs à l'actif

4.2.14 Service d'actif

Un service d'actif sert à intégrer les services d'Enveloppe de Gestion d'Actif qui sont fournis par l'actif proprement dit.

4.2.15 Services relatifs à l'actif

Un service relatif à l'actif est un service qui est logiquement attribué à l'actif, mais qui n'est pas fourni par l'actif proprement dit.

4.3 Aspects du cycle de vie des actifs et des Enveloppes de Gestion d'Actif

Le présent paragraphe 4.3 décrit les caractéristiques du contexte d'application des actifs et des AAS.

Les actifs et les Enveloppes de Gestion d'Actif ont des cycles de vie qu'il convient de considérer séparément. Ceci est expliqué à l'aide de l'exemple d'une machine (voir la Figure 11).

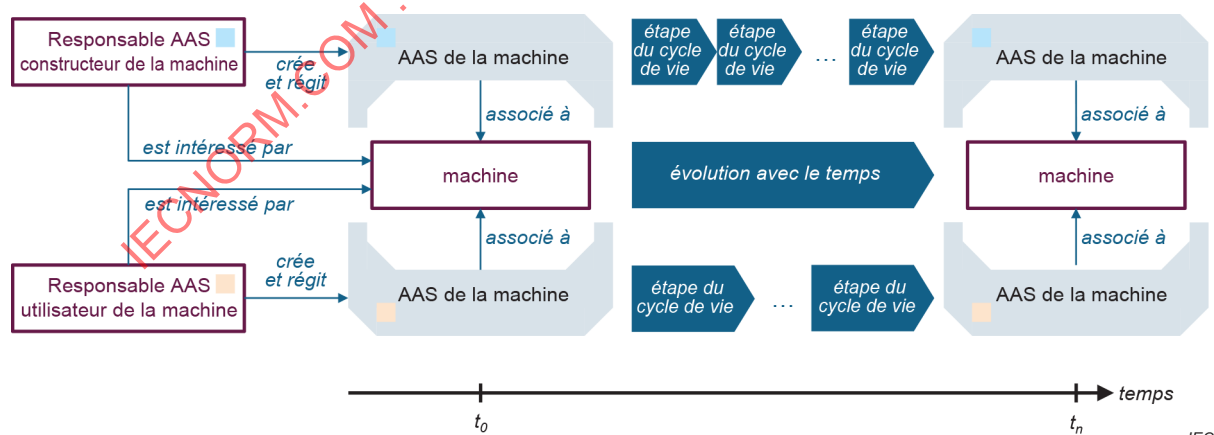


Figure 11 – Exemple d'aspects du cycle de vie des actifs et des Enveloppes de Gestion d'Actif

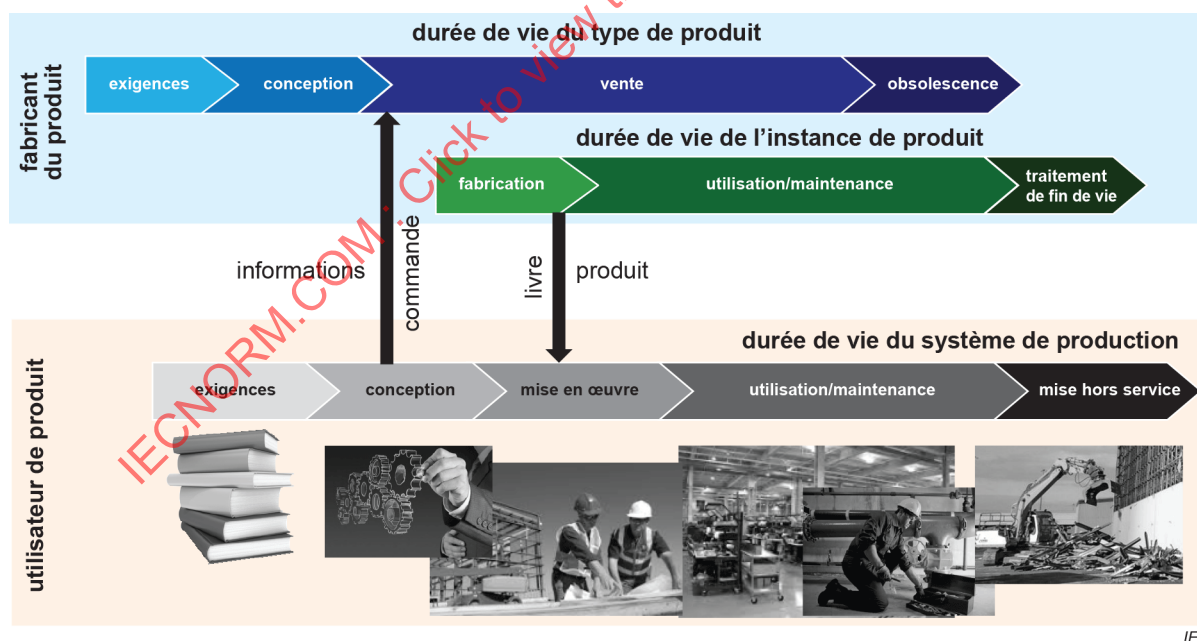
En tant qu'actif physique, une machine change avec le temps, par exemple en raison de l'usure d'un composant, d'une panne, d'une reconfiguration ou de dommages mineurs. Certains de ces changements, comme l'usure, sont intrinsèques à l'actif et peuvent même ne pas être observés par un utilisateur de la machine. D'autres modifications, comme une panne, peuvent être constatées par un utilisateur de la machine, d'autres, comme la reconfiguration, sont généralement initiées par un utilisateur de la machine, et d'autres encore, comme les

dommages mineurs, peuvent même ne pas être pertinents pour l'utilisateur de la machine. Toutes ces modifications sont résumées à la Figure 11 en tant qu'évolution de la machine avec le temps.

Une Enveloppe de Gestion d'Actif change avec le temps, par exemple le responsable d'AAS initie l'ajout d'un Submodel ou d'un service relatif à l'actif, une application utilisateur d'AAS modifie les informations représentées par certains SubmodelElements et un service d'actif fournit des informations à jour sur l'actif résultant en une mise à jour des informations représentées par certains SubmodelElements. Ces modifications ne sont pas intrinsèques, mais résultent des effets explicites des entités relatives à l'Enveloppe de Gestion d'Actif conformément à la Figure 4. Ces modifications sont présentées à la Figure 11 comme des étapes discrètes du cycle de vie. Étant donné que différentes Enveloppes de Gestion d'Actif sont mises en œuvre par différents responsables de l'AAS (rectangles bleu clair et orange clair), ces étapes du cycle de vie sont spécifiques à chaque Enveloppe de Gestion d'Actif.

L'Enveloppe de Gestion d'Actif de l'utilisateur de la machine et celle de l'Enveloppe de Gestion d'Actif du constructeur de machines peuvent être créées à des moments différents, puis généralement coexister. Ces Enveloppes de Gestion d'Actif sont indépendantes et il n'est donc pas garanti que ces Enveloppes de Gestion d'Actif soient logiquement cohérentes. La synchronisation entre ces Enveloppes de Gestion d'Actif dépend des cas d'utilisation sous-jacents et relève de la responsabilité des applications utilisateur d'AAS correspondantes, voir également 4.4.

L'exemple suivant représente les intérêts des différents responsables de l'AAS et leurs points de vue sur les différents actifs et les Enveloppes de Gestion d'Actif associées. Il représente une chaîne de valeur réelle entre le fabricant d'un produit et l'utilisateur d'un produit qui intègre le produit, par exemple un composant d'automatisation, en tant que composant dans le système de production (voir la Figure 12).



IEC

Figure 12 – Exemple d'actifs dans les cycles de vie des produits et des systèmes de production

NOTE L'IEC 62890 définit les types et les instances de produits, l'asynchronisme de leurs cycles de vie, les conséquences sur le cycle de vie du système de production et la dualité du produit et du composant.

Le fabricant du produit (en tant que responsable d'AAS) considère et gère différents actifs au cours de leurs cycles de vie:

- le type de produit qui a un historique avec différentes versions;
- parallèlement, le fabricant produit un certain nombre d'instances de produit du type de produit et de ses versions.

Comme représenté dans l'exemple de la Figure 12, l'instance de produit est représentée par une Enveloppe de Gestion d'Actif d'un actif d'instance, et le type de produit est représenté par une Enveloppe de Gestion d'Actif d'un actif de type.

L'utilisateur du produit incorpore un produit tel qu'un composant dans le système de production. De même, l'utilisateur du produit (en tant que responsable d'AAS) gère les actifs suivants au cours de leurs cycles de vie:

- actif de rôle: l'utilisateur du produit définit le rôle que le produit est destiné à assurer. Ce rôle est inclus dans les descriptions numériques globales du système de production et géré tout au long du cycle de vie du système de production. Ce rôle est également utilisé pour décider de la commande de l'instance de produit correcte auprès du bon fabricant du produit;
- actif d'instance de produit: parallèlement, l'utilisateur du produit gère l'instance de produit (composant) fournie par le fabricant du produit tout au long de son cycle de vie.

Si les rôles sont considérés comme des actifs, ils peuvent être représentés par des Enveloppes de Gestion d'Actif. L'utilisateur du produit représente l'instance de produit (composant) par une Enveloppe de Gestion d'Actif pour un actif d'instance qui comprend des informations fournies par le fabricant du produit et des informations de rôle.

EXEMPLE Au cours de la phase d'exigences, il a été constaté qu'une fonction spécifique pour transporter des pièces vers la machine est nécessaire. La fonction exigée avec ses paramètres (par exemple quantité, poids et taille des pièces) peut être décrite comme un rôle. La description du rôle constitue une valeur en soi, en raison de l'effort exigé pour développer la description et parce que plusieurs experts de différents domaines peuvent être impliqués dans la description. Pour faciliter l'échange d'informations et conserver ces informations pour les phases ultérieures du cycle de vie du système (par exemple pour la maintenance), le gestionnaire d'exigences responsable peut décider de représenter le rôle par une Enveloppe de Gestion d'Actif. La description de rôle constitue ensuite la base de la sélection de la conception optimale pour le transport (par exemple, une bande transporteuse ou un ensemble de véhicules à guidage automatique).

La Figure 13 représente les Enveloppes de Gestion d'Actif pour les actifs représentés à la Figure 12. L'actif "instance de produit" est dérivé de l'actif "type de produit" (voir la Figure 5) et la Figure 13 montre également que cette relation "dérivé de" entre les actifs est représentée par une relation "dérivé de" entre les Enveloppes de Gestion d'Actif, voir 4.2.6.

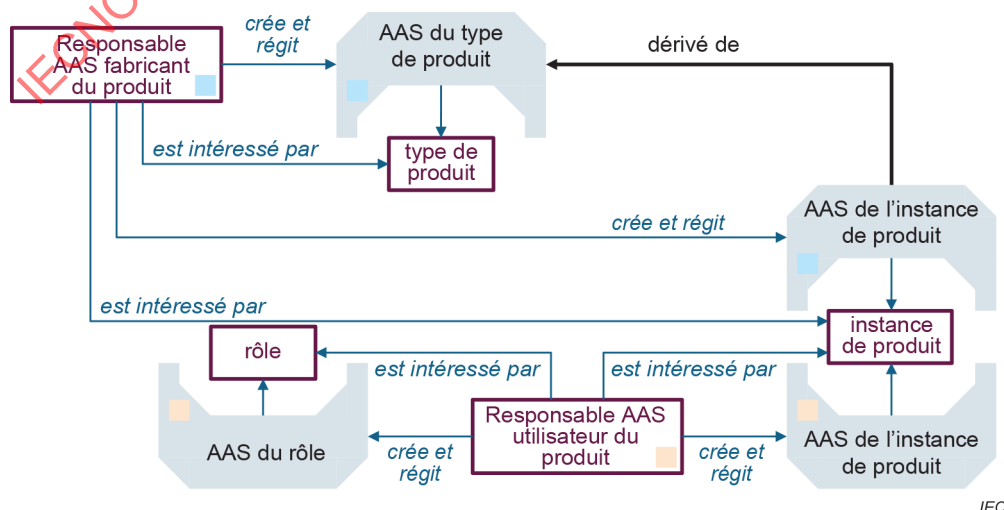


Figure 13 – Exemple d'Enveloppe de Gestion d'Actif dans le cycle de vie du produit

4.4 Exemple d'un scénario global d'Enveloppe de Gestion d'Actif

Le paragraphe 4.4 donne un exemple et intègre les différents points de vue décrits en 4.3 dans un scénario global. La Figure 14 représente la manière dont le point de vue du cycle de vie représenté à la Figure 11, le point de vue d'échange de valeurs représenté à la Figure 12 et le point de vue des actifs et des Enveloppes de Gestion d'Actif représentés à la Figure 13 peuvent être combinés dans un scénario global.

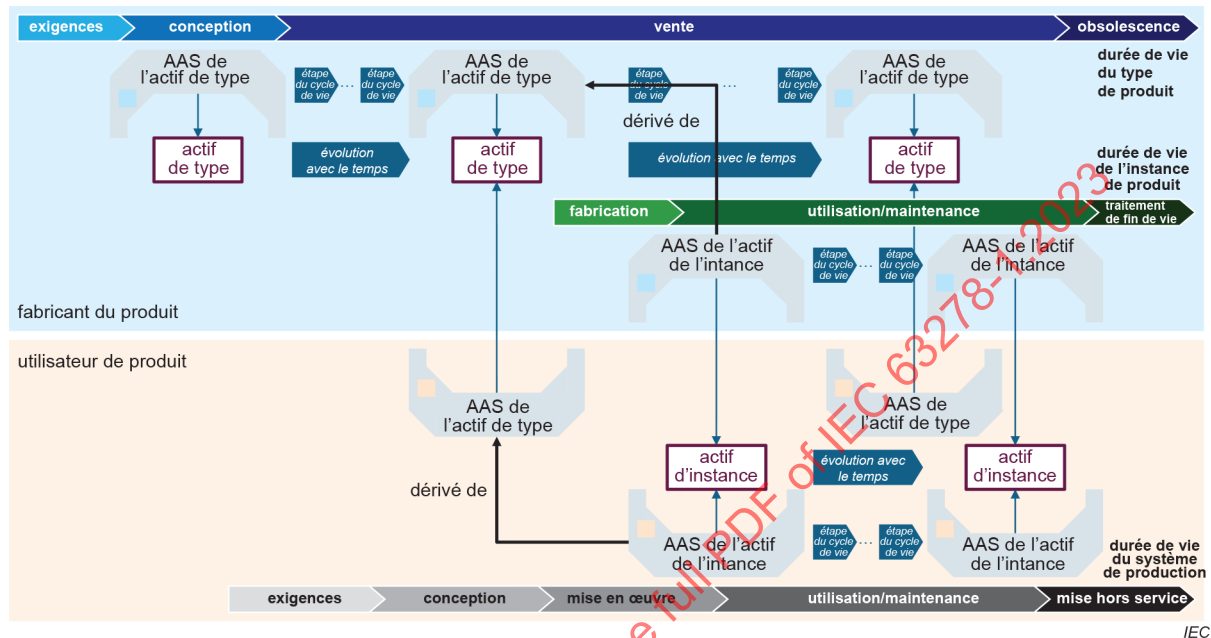


Figure 14 – Représentation d'un exemple de scénario global d'Enveloppe de Gestion d'Actif

La Figure 14 est basée sur les principes de conception suivants:

- la frise chronologique est représentée de gauche à droite;
- pour des raisons de clarté, le responsable d'AAS a été omis, car il est suffisamment exprimé par la coloration en bleu clair et en orange clair;
- la Figure 14 représente deux actifs, l'un est appelé "actif de type" et l'autre "actif d'instance". Deux Enveloppes de Gestion d'Actif sont associées à chaque actif, une Enveloppe de Gestion d'Actif créée et gérée par le fabricant du produit et l'autre créée et gérée par l'utilisateur du produit;
- à un certain moment, l'état correspondant des deux actifs dans leur cycle de vie et la représentation actuelle de l'actif par les Enveloppes de Gestion d'Actif sont représentés. La Figure 14 ne représente pas le fait qu'une infrastructure donnée de l'Enveloppe de Gestion d'Actif peut fournir des capacités pour récupérer la représentation d'un actif à un moment du passé;
- un fabricant du produit pourrait maintenir différentes versions pour un type de produit avec le temps. Cette complexité supplémentaire n'est pas représentée à la Figure 14. Elle peut être modélisée en considérant chaque version comme un actif distinct et en modélisant les relations entre ces différents actifs. Il est présumé que la relation "dérivé de" entre Enveloppes de Gestion d'Actif correspondantes demeure en place tout au long du cycle de vie des Enveloppes de Gestion d'Actif.

La Figure 14 représente un produit qui est intégré dans un système de production de l'utilisateur du produit. L'objectif des différentes Enveloppes de Gestion d'Actif est le suivant:

- enveloppe de Gestion d'Actif pour un actif de type, créée et gérée par le fabricant du produit: le fabricant du produit a généralement intérêt à gérer un type de produit tout au long du cycle de vie;
- enveloppe de Gestion d'Actif pour un actif de type, créée et gérée par l'utilisateur du produit: dans ce scénario, il est présumé que l'utilisateur du produit a un intérêt à gérer le type de produit avec le temps, car l'utilisateur du produit souhaite gérer par exemple ses propres conditions de livraison convenues avec le fabricant du produit, ses propres exigences concernant la commande de plusieurs instances de produit du type de produit ou certains contenus techniques spécifiques concernant le type de produit;
- enveloppe de Gestion d'Actif pour un actif d'instance, créée et gérée par le fabricant du produit: le fabricant du produit a généralement intérêt à gérer chaque instance de produit individuellement tout au long du cycle de vie, en particulier si l'instance de produit est configurée individuellement en fonction des besoins d'un utilisateur de produit. Par exemple, la gestion des configurations spécifiques vendues aux différents utilisateurs de produits ou l'offre de pièces de rechange, qui sont demandées pour une instance de produit spécifique d'un utilisateur de produit;
- enveloppe de Gestion d'Actif pour un actif d'instance, créée et gérée par l'utilisateur du produit: l'utilisateur du produit a généralement intérêt à gérer une instance de produit tout au long du cycle de vie, par exemple pour la gestion de l'entretien de l'instance de produit ou la collecte d'informations d'utilisation de l'instance de produit afin d'optimiser l'utilisation de l'instance de produit. Parfois, un utilisateur de produit ne souhaite pas révéler les informations d'utilisation d'une instance de produit au fabricant du produit.

IECNORM.COM : Click to view the full PDF of IEC 63278-1:2023

La Figure 15 développe la Figure 14 en représentant l'échange de valeurs entre le fabricant du produit et l'utilisateur du produit:

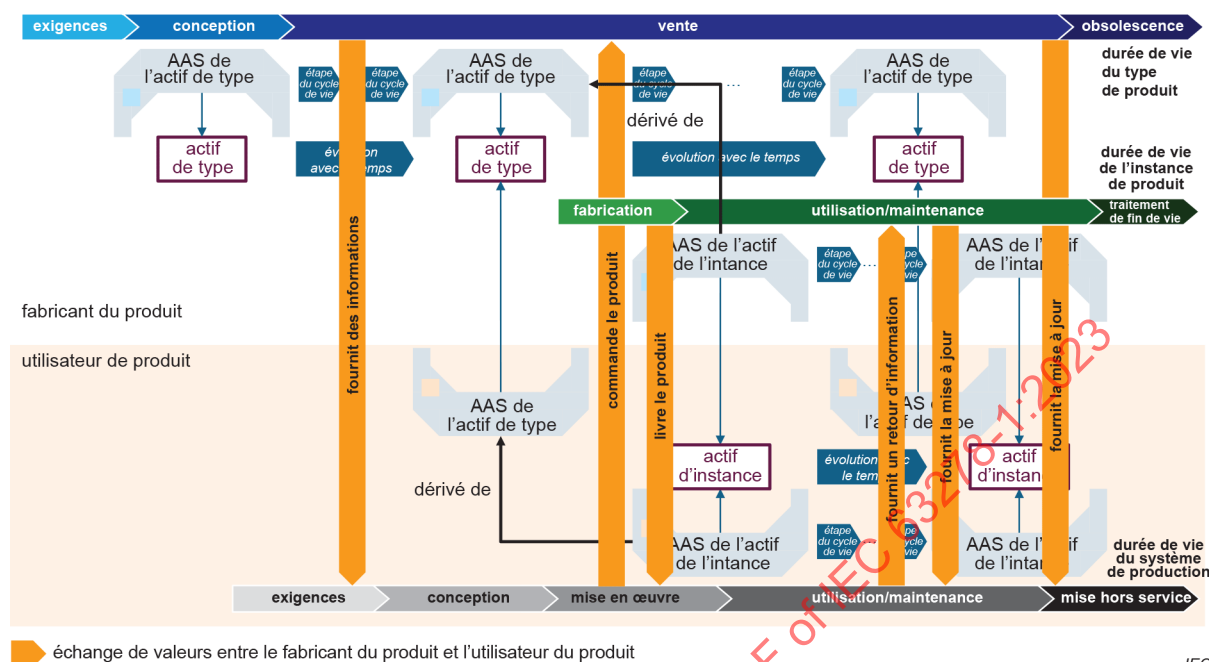


Figure 15 – Représentation de l'échange de valeurs dans le scénario global

- le fabricant du produit fournit des informations sur le type de produit: sur la base de ces informations généralement disponibles représentées dans l'Enveloppe de Gestion d'Actif du fabricant du produit, un utilisateur de produit crée sa propre Enveloppe de Gestion d'Actif du type de produit pour gérer les informations du type de produit pour son propre objectif;
- l'utilisateur du produit commande un produit auprès du fabricant du produit: selon les informations contenues dans l'Enveloppe de Gestion d'Actif du type de produit de l'utilisateur de produit, ce dernier commande un produit spécifique. En général, c'est au plus tard à ce moment-là que l'utilisateur du produit crée une Enveloppe de Gestion d'Actif d'une instance de produit dans laquelle l'utilisateur de produit gère par exemple les informations relatives à la commande. En règle générale, le fabricant du produit crée également au même moment sa propre Enveloppe de Gestion d'Actif de l'instance de produit, dans laquelle le fabricant du produit gère par exemple des informations sur le processus de fabrication de cette instance de produit spécifique;
- le fabricant du produit livre le produit à l'utilisateur du produit: outre le produit physique, le fabricant du produit fournit à l'utilisateur du produit d'autres informations sur l'instance de produit fournie. L'utilisateur du produit prend les informations fournies sur l'Enveloppe de Gestion d'Actif du fabricant du produit et complète l'Enveloppe de Gestion d'Actif de l'instance de produit avec ces informations. Cela montre que le fabricant du produit ne communique pas toutes les informations à l'utilisateur du produit;
- l'utilisateur du produit fournit un retour d'information au fabricant du produit: des exemples de tels retours sont la commande d'une pièce de rechange ou la fourniture d'informations d'utilisation sur le produit afin de tirer parti des services basés sur les données proposés par le fabricant du produit en retour. Le fabricant du produit prend en charge les informations fournies sur l'Enveloppe de Gestion d'Actif de l'utilisateur du produit et complète l'Enveloppe de Gestion d'Actif de l'instance de produit du fabricant du produit avec ces informations. Cela montre que l'utilisateur du produit ne communique pas toutes les informations au fabricant du produit;

- le fabricant du produit fournit des informations actualisées sur l'instance de produit à l'utilisateur du produit: des exemples d'informations actualisées sont la fourniture d'un document mis à jour spécifique à une instance de produit, d'une mise à jour logicielle spécifique à une instance de produit ou d'un service orienté sur les données. L'utilisateur du produit intègre ces informations actualisées dans sa propre Enveloppe de Gestion d'Actif de l'instance de produit. Cela montre que la synchronisation des informations de différentes Enveloppes de Gestion d'Actif d'un même actif dépend des intérêts spécifiques des différents responsables d'AAS des Enveloppes de Gestion d'Actif;
- le fabricant du produit fournit des informations actualisées sur le type de produit: des exemples d'informations actualisées sont la fourniture d'une documentation mise à jour ou de mises à jour logicielles ou de modifications des conditions de livraison. L'utilisateur du produit intègre ces informations actualisées dans sa propre Enveloppe de Gestion d'Actif du type de produit et peut-être également dans sa propre Enveloppe de Gestion d'Actif de l'instance de produit.

5 Identificateurs

5.1 Besoins

L'Enveloppe de Gestion d'Actif et ses éléments peuvent être accessibles par des applications utilisateur d'AAS. Ces applications utilisateur d'AAS sont créées et utilisées par différentes organisations du monde entier (voir 4.2.1, 4.2.3).

Les SubmodelElements avec une relation d'objectif peuvent être utilisés pour exprimer les relations entre ces Enveloppes de Gestion d'Actif et leurs éléments (voir 4.2.9). Les éléments suivants exigent des identificateurs uniques: Enveloppe de Gestion d'Actifs, Submodels, actifs, concepts utilisés, responsable d'AAS, application utilisateur d'AAS, modèles Submodel (voir 4.2). Pris conjointement avec l'espace de noms global, ces identificateurs doivent être uniques au niveau global.

NOTE D'autres éléments, tels que les SubmodelElements (voir 4.2.9), sont structurés de manière hiérarchique par d'autres éléments, comme les Submodels, et n'exigent donc pas d'identificateur distinct au niveau global.

EXEMPLE Une machine est qualifiée pour l'industrie alimentaire. Elle renferme plusieurs entraînements provenant de différents fabricants de composants. Le constructeur de la machine et les fabricants de composants font partie du marché mondial. Afin de procéder à la qualification, les Submodels pour les données techniques de chacun de ces entraînements sont accessibles et évalués pour vérifier si la lubrification est conforme aux réglementations de l'industrie alimentaire.

5.2 Détermination des identificateurs

5.2.1 Généralités

Pour les normes et spécifications internationales et librement disponibles ci-après, des variantes d'identificateurs appropriées sont déterminées (voir la Figure 16).

Il existe deux catégories d'identificateurs pour lesquelles les exigences sont différentes:

- les identificateurs, qui ont été fournis par des rôles externes et qui sont simplement à utiliser, tels que des identificateurs distincts au niveau global pour les entrées des répertoires de concepts;
- les identificateurs, qui sont à créer par le responsable d'AAS (ou l'application utilisateur d'AAS respective), tels que des identificateurs distincts au niveau global pour actifs, Enveloppes de Gestion d'Actif et Submodels.

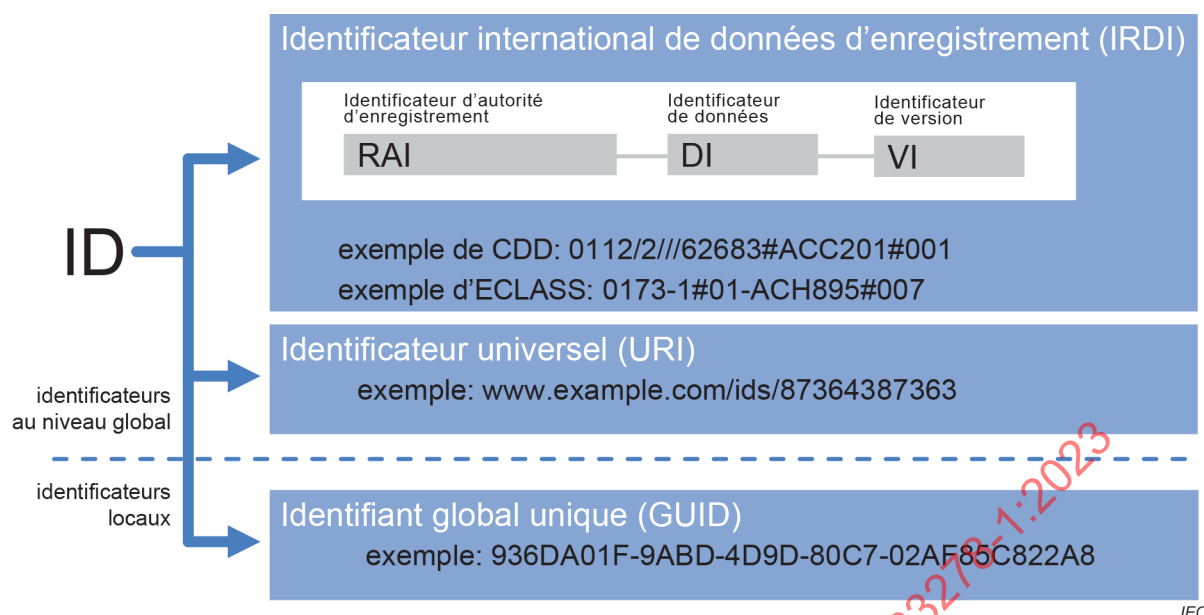


Figure 16 – Différents identificateurs pour les identificateurs distincts au niveau global et les identificateurs locaux

5.2.2 Identificateurs distincts au niveau global pour les concepts par IRDI

L'ISO/IEC 11179-6, l'ISO/IEC 6523 et l'ISO 29002 établissent un schéma d'identification pour des identificateurs distincts au niveau global, appelés IRDI. Le caractère unique d'un IRDI est déterminé par la combinaison des valeurs de trois composants:

- identificateur d'autorité d'enregistrement (RAI);
- identificateur de données (DI);
- identificateur de version (VI).

5.2.3 Identificateurs distincts au niveau global par URI

Au moyen du schéma URI (NETF RFC 3986 [3]), des volumes d'identificateurs de domaine et de chemin, sûrs de type, distincts et distribués au niveau global peuvent être réalisés.

Les URI conviennent comme identificateurs pour les entrées de répertoire de concepts, les actifs, les Enveloppes de Gestion d'Actif, les Submodels, les responsables d'AAS et les applications utilisateur d'AAS.

EXEMPLE 1 Une organisation crée des identificateurs par adresse universelle (URL). Les identificateurs ont le nom d'hôte qui est enregistré pour cette organisation. Une procédure d'énumération des chemins qui suivent le nom d'hôte au sein de l'organisation permet de s'assurer que seuls les chemins uniques sont énumérés pour des identificateurs particuliers. La prise conjointe du nom d'hôte et de chemins assure des identificateurs distincts au niveau global par URL.

EXEMPLE 2 Une organisation crée des identificateurs par adresse URN. Un identifiant d'espace de nom (NID) distinct est choisi pour les URN, par exemple en fonction du nom d'hôte de l'organisation. Pour le Namespace Specific String (NSS), une procédure d'énumération est établie comme ci-dessus. La prise conjointe du NID et du NSS assure des identificateurs distincts au niveau global par adresse URN.

5.2.4 Identificateurs locaux

Les identificateurs locaux sont appropriés s'ils ne sont pas accessibles par les applications utilisateur d'AAS utilisées par des organisations autres que le responsable d'AAS. Par exemple, ils peuvent être des identifiants globaux uniques (GUID) ou des identificateurs des éléments de données internes du fabricant.

Cependant, pour les identificateurs des éléments de données internes des fabricants, il est recommandé d'utiliser des IRDI, car ils sont uniques lorsqu'un indicatif de code international (ICD, International Code Designator) spécifique à l'entreprise est attribué.

6 Structure de l'Enveloppe de Gestion d'Actif

6.1 Généralités

Dans l'Article 6, une structure générale pour une Enveloppe de Gestion d'Actif est définie en indiquant des exigences, des recommandations et des autorisations. L'accent est mis sur les parties grises représentées à la Figure 4. Si une mise en œuvre de l'Enveloppe de Gestion d'Actif satisfait à ces exigences, recommandations et autorisations, l'utilisation de l'Enveloppe de Gestion d'Actif décrite à l'Article 4 est possible et l'interopérabilité est obtenue.

NOTE Pour une discussion sur l'interopérabilité, voir 0.3.

La structure générale est basée sur des travaux conceptuels établis dans les technologies d'automatisation et TIC. La structure définie est préparée pour les développements futurs concernant les aspects pertinents tels que l'intégration horizontale et l'intégration verticale. L'objectif de la Figure 17 est de démontrer la structure avec un exemple.

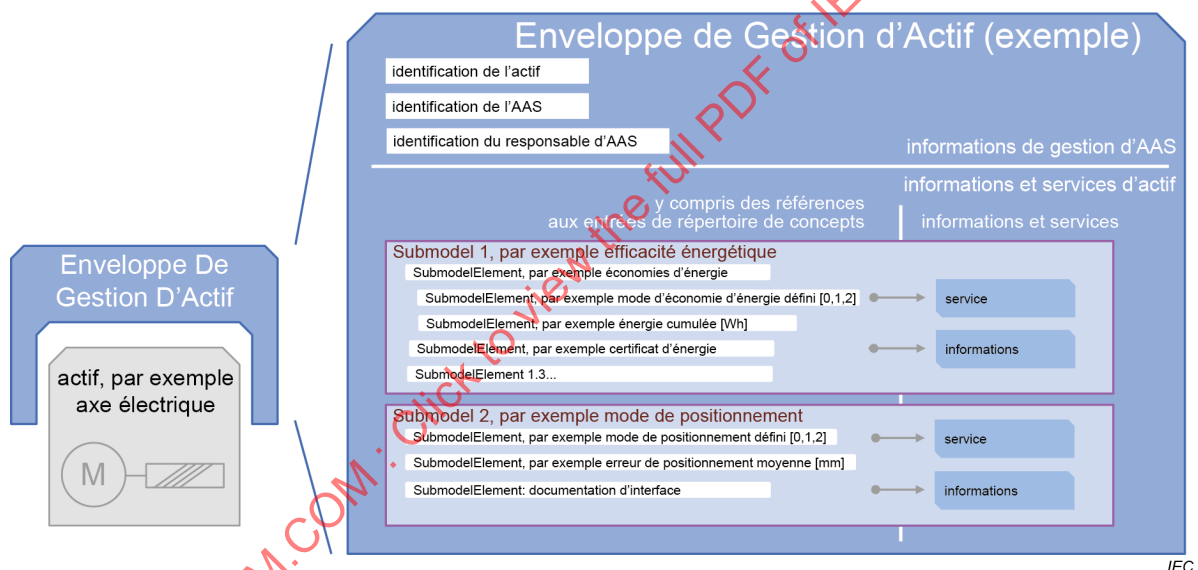


Figure 17 – Exemple d'une Enveloppe de Gestion d'Actif démontrant la structure générale

Les informations de gestion d'AAS de la Figure 17 sont utilisées pour identifier les Enveloppes de Gestion d'Actif, l'actif associé et le responsable d'AAS associé, ainsi que pour gérer l'Enveloppe de Gestion d'Actif tout au long de son cycle de vie (voir 4.3).

NOTE Selon le type d'actif, qui est représenté par l'Enveloppe de Gestion d'Actif, les informations de gestion d'AAS sont similaires à l'en-tête de classe d'actifs DF ou à l'en-tête d'actif DF de la série IEC 62832. Dans l'IEC 62832, un en-tête fournit au moins l'identificateur, le numéro de version, le nom préférentiel et la description de l'élément. Les informations de gestion d'AAS fournissent des informations supplémentaires spécifiques à l'Enveloppe de Gestion d'Actif.

Les informations et services d'actif de la Figure 17 comprennent des Submodels individuels. Comme représenté à la Figure 17, chaque Submodel a des SubmodelElements organisés hiérarchiquement, chacun faisant référence à des informations et services individuels (représentés par les rectangles bleus à la Figure 17). Les services peuvent être des services d'actif (par l'intermédiaire de l'intégration d'actif) ou des services relatifs à l'actif.

Les informations de gestion d'AAS et les informations et services d'actif sont protégés par des dispositions relatives à la sécurité de l'information appropriées pour l'application industrielle des actifs concernés (voir 6.3).

6.2 Exigences associées à l'Enveloppe de Gestion d'Actif

6.2.1 Généralités

Les dispositions du paragraphe 6.2 sont dérivées de l'analyse du cadre conceptuel de l'Article 4. Les dispositions sont séparées dans les paragraphes suivis d'autres énoncés, notes et exemples.

6.2.2 Enveloppe de Gestion d'Actif

6.2.2.1 Exigences relatives à la gestion de l'information

6.2.2.1.1 Exigences

L'Enveloppe de Gestion d'Actif doit avoir un identificateur distinct au niveau global. Pour l'identificateur, l'URI doit être utilisée.

Une Enveloppe de Gestion d'Actif doit avoir un identificateur distinct au niveau global de l'actif qu'elle représente. Pour l'identificateur, l'URI doit être utilisée.

Il convient qu'une Enveloppe de Gestion d'Actif ait un identificateur distinct pour le responsable d'AAS.

Une Enveloppe de Gestion d'Actif doit fournir un ensemble défini d'informations de gestion d'AAS.

Une Enveloppe de Gestion d'Actif doit prendre en charge la gestion des informations de gestion d'AAS tout au long du cycle de vie.

6.2.2.1.2 Justification et recommandations

L'identificateur de l'Enveloppe de Gestion d'Actif est différent de l'identificateur de l'actif. L'Enveloppe de Gestion d'Actif doit être identifiée de manière unique, car plusieurs responsables d'AAS peuvent créer différentes Enveloppes de Gestion d'Actif pour le même actif (voir la Figure 6).

NOTE 1 Selon le type d'actif représenté, l'identificateur de l'Enveloppe de Gestion d'Actif est équivalent à l'identificateur d'élément d'une classe d'actifs DF (type d'actif = type) ou à l'identificateur d'élément d'un actif DF (type d'actif = instance) conformément à la série IEC 62832.

NOTE 2 Il est possible d'utiliser un GUID et d'exprimer ceci comme une URI.

L'association entre les actifs et les Enveloppes de Gestion d'Actif est très importante. Ces associations ne doivent pas se briser, même si elles sont sauvegardées dans des répertoires numériques ou sauvegardées d'une manière qui couvre tous les partenaires à valeur ajoutée.

NOTE 3 L'identificateur unique de l'actif peut être mis à disposition par l'actif proprement dit, par exemple sur sa plaque signalétique.

NOTE 4 L'identificateur de l'actif est équivalent à l'identificateur d'actif PS de l'actif DF conformément à la série IEC 62832.

NOTE 5 L'utilisation d'URI comme identificateurs d'actifs est décrite dans l'IEC 61406-1:2022 et dans l'IEC 61406-2 CDV.

L'identificateur du responsable d'AAS est destiné à reconnaître qui est responsable d'une Enveloppe de Gestion d'Actif spécifique.

Les dispositions relatives à la gestion de l'information sont nécessaires pour gérer une Enveloppe de Gestion d'Actif tout au long de son cycle de vie.

NOTE 6 Les informations de gestion d'AAS sont par exemple l'identificateur d'actif distinct au niveau global, les informations de version, les horodatages de données ou de modifications structurelles, etc.

6.2.2.2 Exigences relatives à la structure

6.2.2.2.1 Exigences

Une Enveloppe de Gestion d'Actif doit comporter des Submodels comprenant des SubmodelElements et une ou plusieurs interfaces d'AAS.

6.2.2.2.2 Justification et recommandations

Pour prendre en charge l'interopérabilité entre plusieurs applications utilisateur d'AAS, il est important de disposer d'une structure bien connue d'informations et d'interfaces d'AAS.

6.2.2.3 Exigences relatives à la classification d'actifs

6.2.2.3.1 Exigences

Une Enveloppe de Gestion d'Actif doit décrire si un actif est considéré comme un actif de type ou comme un actif d'instance ou si ce type de classification n'est pas applicable à l'actif.

Un actif d'instance doit être représenté par une Enveloppe de Gestion d'Actif pour un actif d'instance, et un actif de type doit être représenté par une Enveloppe de Gestion d'Actif pour un actif de type.

6.2.2.3.2 Justification et recommandations

La Figure 5 et l'explication de 4.2.3 fournissent une justification et des recommandations relatives à la classification d'actifs.

6.2.3 Submodel

6.2.3.1 Exigences générales

6.2.3.1.1 Exigences

Une Enveloppe de Gestion d'Actif doit prendre en charge les Submodels en fonction des besoins du responsable d'AAS.

Une Enveloppe de Gestion d'Actif doit prendre en charge la gestion d'un Submodel tout au long de son cycle de vie.

6.2.3.1.2 Justification et recommandations

Les Submodels sont utilisés pour structurer les informations et les services d'une Enveloppe de Gestion d'Actif en parties pouvant clairement être distinguées. Chaque Submodel fait référence à un domaine ou à un sujet bien défini.

Les actifs évoluant avec le temps, les représentations numériques doivent être maintenues tout au long de la durée de vie (voir 4.3 et 4.4).

6.2.3.2 Exigences relatives à la structure

6.2.3.2.1 Exigences

Un Submodel doit prendre en charge l'organisation de SubmodelElements en structures hiérarchiques.

6.2.3.2.2 Justification et recommandations

Les Submodels organisent des SubmodelElements. Pour faire face au volume de SubmodelElements, un moyen de structuration hiérarchique est exigé.

NOTE 1 Un Submodel, qui comporte des SubmodelElements tels que des propriétés, correspond à un CDEL selon la série IEC 62832 et peut avoir une sous-structure hiérarchique. Une définition de CDEL peut décrire la sémantique d'un Submodel.

NOTE 2 Un ensemble de SubmodelElements correspond à un CDEL de la série IEC 62832 pour la structuration hiérarchique.

6.2.3.3 Exigences relatives à la sémantique

6.2.3.3.1 Exigences

Il convient que le concept d'Enveloppe de Gestion d'Actif prenne en charge la création de Submodels guidés par des modèles Submodel prédéfinis, en plus des Submodels, où ses SubmodelElements font référence aux entrées de répertoire de concepts des répertoires de concepts.

S'il est guidé par un modèle Submodel, le Submodel doit faire référence au modèle Submodel par son identificateur distinct au niveau global.

6.2.3.3.2 Justification et recommandations

Les modèles Submodel servent à guider un Submodel en faisant en sorte que ce Submodel dispose d'un ensemble garanti de SubmodelElements. Les modèles Submodel ont donc leur propre cycle de vie, même indépendant du cycle de vie des répertoires de concepts de références, car le propriétaire du modèle Submodel peut être différent du responsable d'AAS et du fournisseur des répertoires de concepts.

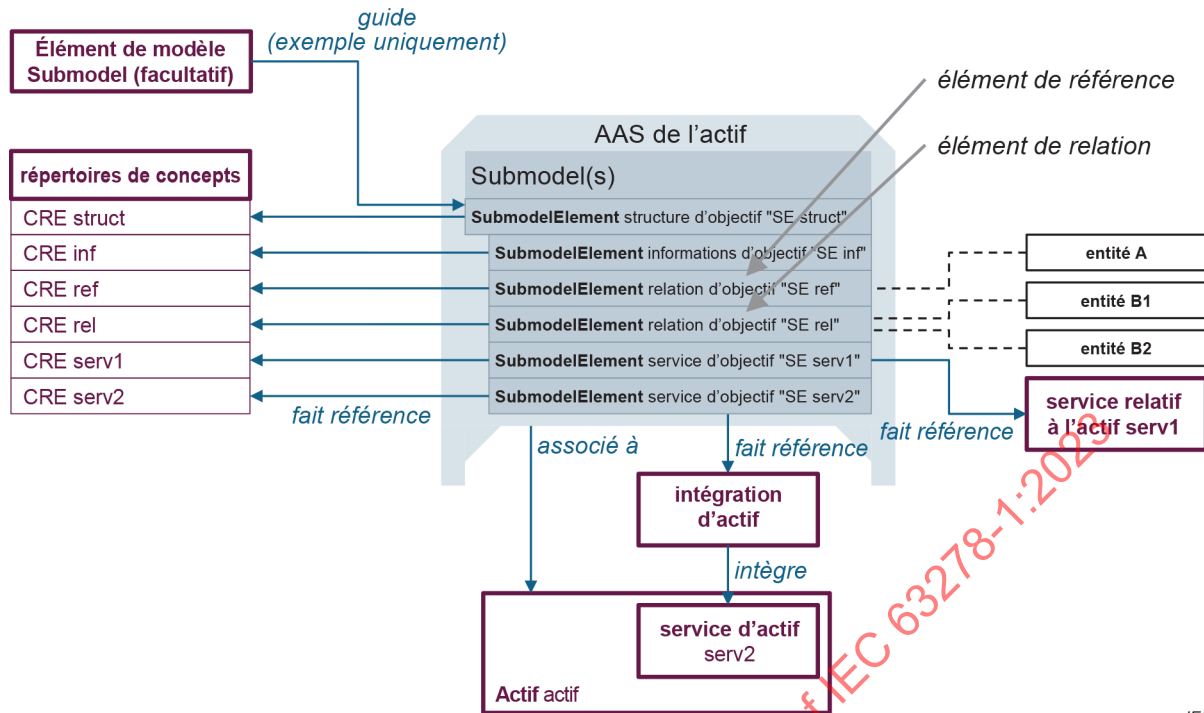
NOTE 1 Ces modèles Submodel peuvent normaliser les informations et les services, par exemple pour une plaque signalétique numérique des actifs, les fiches techniques des composants ou la documentation de réception technique des équipements.

NOTE 2 Des documents supplémentaires peuvent définir comment les modèles Submodel sont normalisés dans la normalisation internationale.

6.2.4 SubmodelElements

6.2.4.1 Généralités

Conformément à la Figure 4, les différents aspects des SubmodelElements sont représentés à la Figure 18 pour servir de recommandations pour les différentes exigences relatives aux SubmodelElements.



IEC

Figure 18 – Représentation des différents aspects des SubmodelElements

6.2.4.2 Exigences générales

6.2.4.2.1 Exigences

Un SubmodelElement doit prendre en charge la description des informations.

Un SubmodelElement doit prendre en charge la description de la référence à une entité clairement identifiée.

Un SubmodelElement doit prendre en charge la description d'une relation entre deux entités clairement identifiées.

Un SubmodelElement doit prendre en charge la description de la référence à un service d'actif par l'intermédiaire d'une intégration d'actif.

Un SubmodelElement doit prendre en charge la description de la référence à un service relatif à l'actif.

Un SubmodelElement doit prendre en charge l'organisation de SubmodelElements en structures hiérarchiques.

6.2.4.2.2 Justification et recommandations

Selon la Figure 17, la structure des Submodels et des SubmodelElements dans une Enveloppe de Gestion d'Actif constitue une table de matières clairement définie pour toutes les informations et tous les services. Les SubmodelElements sont de structure uniforme et fournissent ainsi une source d'informations très stable.

Dans le domaine de la fabrication, les informations et les services présentent une forte variance et ont souvent une structure très complexe. Par conséquent, les SubmodelElements se réfèrent à ces données, modèles numériques et services complexes afin de les exposer à la ou aux interfaces d'AAS dans une approche uniforme.

NOTE 1 Cette approche repose sur une compréhension approfondie du concept de propriété de la série IEC 61360.

Les informations et les services ont souvent des relations avec d'autres entités. Pour pouvoir parcourir ces relations en toute sécurité, il est nécessaire d'identifier clairement les entités référencées. Cela inclut, sans prétendre à l'exhaustivité, par exemple: l'identificateur de l'entité (voir l'Article 5), les informations relatives au type d'entité (par exemple un actif, une Enveloppe de Gestion d'Actif, un Submodel ou un SubmodelElement) ou l'identificateur d'une Enveloppe de Gestion d'Actif, si des entités d'une autre Enveloppe de Gestion d'Actif sont à référencer. Ces informations peuvent être décrites par des SubmodelElements appropriés. Les détails sont définis dans l'IEC 63278-2.

NOTE 2 Pour référencer les SubmodelElements, un identificateur unique d'un SubmodelElement peut être obtenu en utilisant un identificateur unique d'un Submodel enchaîné avec des identificateurs de SubmodelElements subordonnés et un identificateur du SubmodelElement proprement dit.

Une relation entre deux SubmodelElements combine deux références avec un identificateur sémantique pour une entrée de répertoire de concept.

6.2.4.2.3 Exemples d'exigences

Les SubmodelElements exigent la représentation des caractéristiques des actifs pour plusieurs raisons. Le Tableau 1 fournit des exemples pour les objectifs pris en charge par les SubmodelElements.

Tableau 1 – Exemples de catégorisation des SubmodelElements

Spécialisation de SubmodelElement	Description	Objectif
Ensemble de SubmodelElements	Représente les structures hiérarchiques, les jeux et les listes de SubmodelElements.	Structure
Instance de propriété	Représente les caractéristiques faisant référence aux types de propriétés, par exemple dans les dictionnaires normalisés.	Informations
Fichier	Représente des informations complexes d'un format de données spécifique relatif à une entrée de répertoire de concepts unique.	Informations
Élément de référence	Représente une référence compréhensible à une autre entité, telle qu'un actif, une Enveloppe de Gestion d'Actif, un Submodel ou un SubmodelElement.	Relation
Élément de relation	Représente une combinaison triple de connaissances: sujet, prédicat et objet. Le sujet et l'objet sont décrits par des références. Le prédicat est décrit par l'élément proprement dit et son entrée de répertoire de concepts.	Relation
Élément de service	Représente un service et permet l'exécution d'un service, par exemple des services d'actif ou des services relatifs à l'actif.	Service

NOTE 1 La représentation des instances de propriétés conformément à la série IEC 61360 (voir l'Article A.2) et à la série ISO 22745 (OTD) est basée sur le format de DataElement défini par l'IEC 62832-2. Pour différents types d'informations et de fonctions, différents formats de données complémentaires et méthodes d'accès sont utilisés.

L'une des relations types utilisées pour décrire un actif composite est la relation "composé de", voir la Figure 5. La Figure 19 illustre un exemple de relation "composé de" d'un ensemble d'axes électriques.

EXEMPLE Un ensemble d'axes mécaniques et de préhenseurs constitue un ensemble encapsulé, représenté par une Enveloppe de Gestion d'Actif dédiée et fait référence à ses Enveloppes de Gestion d'Actif subordonnées pour les axes mécaniques et préhenseurs individuels. Chaque Enveloppe de Gestion d'Actif expose une interface d'AAS individuelle à utiliser par les applications utilisateur d'AAS.

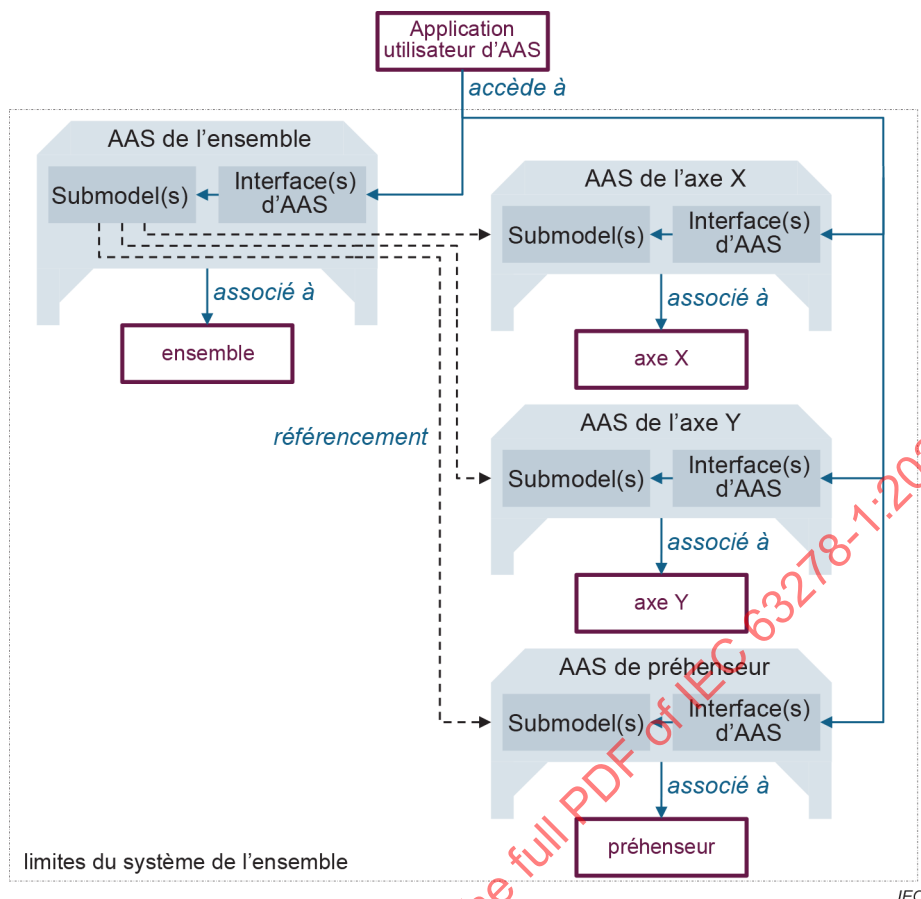


Figure 19 – Envelopes de Gestion d'Actif d'un ensemble représentatif d'axes électriques

Une autre relation typique utilisée pour décrire une relation entre une Enveloppe de Gestion d'Actif représentant un actif d'instance et l'Enveloppe de Gestion d'Actif représentant l'actif de type correspondant est la relation "dérivé de". La relation "peut être dérivé de" entre l'instance et l'actif de type, représentée à la Figure 5 et à la Figure 13 comme une relation entre les Envelopes de Gestion d'Actif correspondantes, est représentée par un SubmodelElement avec une relation d'objectif faisant référence à une entrée du répertoire de concepts "dérivé de".

En règle générale, les Submodels sélectionnés de l'Enveloppe de Gestion d'Actif qui représentent l'actif de type sont pris en charge par l'Enveloppe de Gestion d'Actif qui représente l'actif d'instance. Cependant, une Enveloppe de Gestion d'Actif représentant un actif d'instance peut comporter des Submodels supplémentaires et peut également modifier les valeurs des SubmodelElements pris en charge.

NOTE 2 Par exemple, cela peut également signifier que les représentations d'un actif de type sont étendues tout au long du cycle de vie ou, pour l'Enveloppe de Gestion d'Actif d'un actif d'instance, des Submodels ou des SubmodelElements sont ajoutés, modifiés ou supprimés en fonction des activités (maintenance) de l'actif correspondant.

6.2.4.3 Exigences relatives à la sémantique

6.2.4.3.1 Exigences

Un SubmodelElement doit prendre en charge la spécification d'une référence à une entrée de répertoire de concepts fournie par un répertoire de concepts.

Pour les identificateurs des entrées de répertoire de concepts, tout type d'identificateur qui fournit un identificateur distinct au niveau global doit être pris en charge.

Une Enveloppe de Gestion d'Actif doit prendre en charge différents répertoires de concepts devant être référencés par différents SubmodelElements.

Il convient que le concept d'Enveloppe de Gestion d'Actif prenne en charge la création d'un SubmodelElement guidé par un élément de modèle Submodel prédéfini.

6.2.4.3.2 Justification et recommandations

Les références aux répertoires de concepts bien connus sont importantes pour l'interopérabilité sémantique.

NOTE 1 Les dictionnaires de données bien connus tels que l'IEC CDD, l'ODD ou l'ECLASS utilisent un IRDI.

NOTE 2 Pour les ontologies trouvées sur Internet, des identificateurs conformes à l'URI sont utilisés.

La nécessité de différents répertoires de concepts est expliquée en 4.2.12.

Les recommandations des éléments des modèles Submodel suivent les mêmes principes que les recommandations des modèles Submodel (voir 4.2.10).

6.2.5 Interface d'AAS

6.2.5.1 Exigences

Toutes les Enveloppes de Gestion d'Actif doivent fournir un jeu d'interfaces d'AAS normalisées pour permettre aux applications utilisateur d'AAS d'accéder aux informations et d'accéder et d'exécuter les services qu'ils offrent de manière uniforme.

Toutes les Enveloppes de Gestion d'Actif doivent fournir un jeu d'interfaces d'AAS normalisées pour permettre aux applications utilisateur d'AAS de créer, de supprimer et d'actualiser une Enveloppe de Gestion d'Actif et les Submodels et SubmodelElements de manière uniforme.

Toutes les Enveloppes de Gestion d'Actif doivent fournir un jeu d'interfaces d'AAS normalisées pour permettre aux applications utilisateur d'AAS de récupérer une Enveloppe de Gestion d'Actif non seulement par l'identificateur unique de l'actif qu'elle représente, mais également par d'autres identificateurs du même actif. Ces autres identificateurs ne doivent être uniques que dans un espace de noms donné, mais pas au niveau global.

6.2.5.2 Justification et recommandations

Une interface d'AAS rend les informations et les services de l'Enveloppe de Gestion d'Actif disponibles depuis l'extérieur. Certains exemples sont (la liste n'est pas exhaustive):

- lire/écrire des SubmodelElements avec informations d'objectif;
- exécuter des SubmodelElements avec service d'objectif;
- explorer des Submodels et des SubmodelElements avec une structure et une relation d'objectif;
- créer, supprimer et mettre à jour pour une Enveloppe de Gestion d'Actif, un Submodel et un SubmodelElement;
- récupérer une Enveloppe de Gestion d'Actif.

NOTE 1 L'interface d'AAS normalisée sera définie dans une partie future de la série IEC 63278.

NOTE 2 Conformément à ces concepts de base fournis par les interfaces d'AAS, une application utilisateur d'AAS peut mettre en œuvre des concepts plus complexes tels que la copie, l'écriture miroir et le versionnage. Certains de ces concepts complexes sont représentés en 4.4.

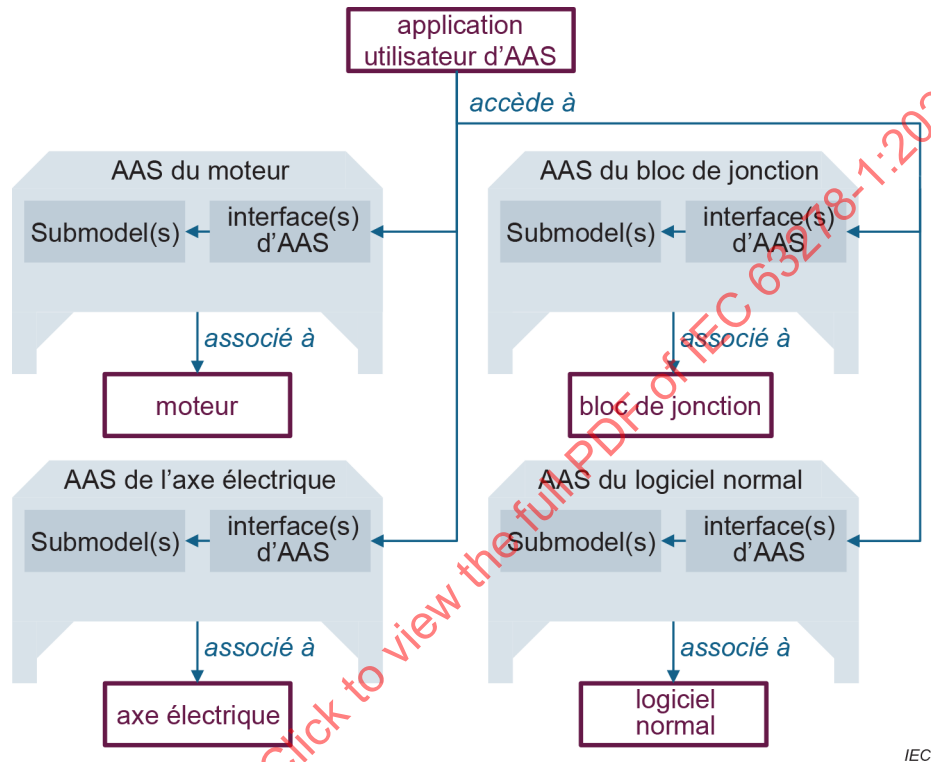
NOTE 3 La ou les interfaces d'AAS peuvent être réalisées par une interface de programmation d'application (API). Un exemple de mise en œuvre d'une ou de plusieurs interfaces d'AAS est l'architecture unifiée de l'OPC (voir l'Article A.7).

Dans les applications industrielles, plusieurs identificateurs sont utilisés pour récupérer des informations d'un actif par des applications logicielles. Tous ces identificateurs ne sont pas distincts au niveau global, ils sont définis par le contexte et sont uniques dans un espace de noms donné.

NOTE 4 Un tel autre identificateur peut être le code article international.

6.2.5.3 Exemple d'exigence

Le paragraphe 6.2.5.3 et la Figure 20 décrivent un exemple des exigences.



EXEMPLE Une Enveloppe de Gestion d'Actif représente les informations et les services associés à l'actif correspondant. L'idée de l'Enveloppe de Gestion d'Actif facilite le maintien de la représentation numérique correspondant à l'actif réel. Dans l'exemple, pour un ensemble d'actif, des Enveloppes de Gestion d'Actif correspondantes sont données.

Pour accéder à ces représentations numériques, une interface normalisée est utilisée. Cette interface peut également être utilisée pour une conservation permanente des informations dans les Enveloppes de Gestion d'Actif correspondantes et peut fournir des options de récupération efficaces. Une application utilisateur d'AAS peut accéder à cette interface (voir 4.2.7).

Figure 20 – Application utilisateur d'AAS accédant aux interfaces d'AAS des Enveloppes de Gestion d'Actif

6.3 Exigences relatives à l'exposition des données et à la sécurité de l'information

6.3.1 Exposition des données d'une Enveloppe de Gestion d'Actif et de ses Submodels

6.3.1.1 Exigences

Une Enveloppe de Gestion d'Actif doit prendre en charge le contrôle de l'exposition des données des informations de gestion d'AAS et des Submodels sur la base d'éléments individuels, conformément à la gouvernance du responsable d'AAS vis-à-vis de chaque application utilisateur d'AAS.

6.3.1.2 Justification

Le responsable d'Enveloppe de Gestion d'Actif ne souhaite pas que toutes les informations incluses dans l'Enveloppe de Gestion d'Actif ou le Submodel, ou les deux, soient exposées à toutes les applications utilisateur d'AAS, mais il souhaite contrôler l'exposition de ces informations conformément aux stratégies métier, personnelles ou autres du responsable d'AAS vis-à-vis de chaque application utilisateur d'AAS.

La définition des contrôles de sécurité appropriés constitue l'un des moyens de contrôler l'exposition de l'Enveloppe de Gestion d'Actif et de ses Submodels.

6.3.2 Exigences générales relatives à la sécurité de l'information de l'Enveloppe de Gestion d'Actif

6.3.2.1 Exigence

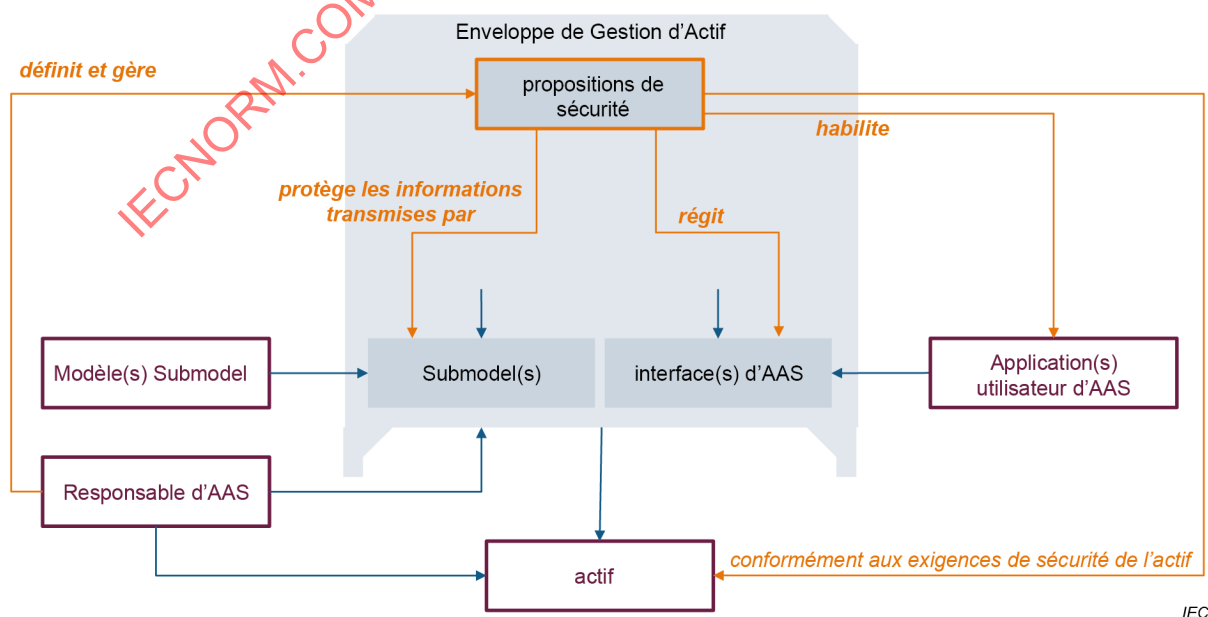
Une Enveloppe de Gestion d'Actif doit prendre en charge la modélisation des propositions de sécurité relatives à la disponibilité, à l'intégrité et à la confidentialité pour les Submodels et la ou les interfaces d'AAS.

6.3.2.2 Justification et recommandations

L'objectif de la sécurité de l'Enveloppe de Gestion d'Actif est de protéger la disponibilité, l'intégrité et la confidentialité des informations et des services fournis par une Enveloppe de Gestion d'Actif conformément aux règles définies par le responsable d'AAS. Outre les SubmodelElements contenus dans les Submodels individuels, l'Enveloppe de Gestion d'Actif doit fournir des moyens de fournir les informations exigées pour la gouvernance de la sécurité.

Outre les informations et les services fournis par l'Enveloppe de Gestion d'Actif, la sécurité de l'Enveloppe de Gestion d'Actif doit également tenir compte de la sécurité de l'actif associé, de l'application utilisateur d'AAS et du modèle Submodel conformément à la Figure 2. En particulier, tout impact négatif de l'Enveloppe de Gestion d'Actif concernant la sécurité des TI et TO doit être empêché.

La Figure 21 représente une extension de la Figure 2 concernant les aspects relatifs à la sécurité de l'Enveloppe de Gestion d'Actif.



IEC

Figure 21 – Vue d'ensemble de la sécurité de l'Enveloppe de Gestion d'Actif

Les propositions de sécurité décrivent la manière dont les informations et services spécifiques fournis par les Submodels sont traités en fonction de la disponibilité, de l'intégrité et de la confidentialité des objectifs de protection. Les propositions de sécurité peuvent être réalisées sous forme de SubmodelElements supplémentaires d'un Submodel existant ou de Submodels de sécurité dédiés distincts qui se rapportent aux Submodels existants et les complètent.

Le responsable d'AAS définit et gère les propositions de sécurité applicables aux informations et services fournis avec ou par les Submodels. Selon un scénario d'utilisation spécifique, un ensemble différent de qualités de disponibilité, d'intégrité et de confidentialité peut s'appliquer, par exemple le responsable d'AAS peut décider des éléments d'un Submodel mis à la disposition d'une application utilisateur d'AAS (confidentialité) spécifique et des dispositions qui s'appliquent en ce qui concerne l'intégrité et la disponibilité (QoS) des informations et services fournis. L'interface d'AAS contrôle l'exposition des informations fournies par les Submodels vers l'application utilisateur d'AAS conformément aux propositions de sécurité.

Les applications utilisateur d'AAS autorisées ont un droit d'accès aux informations et aux services désignés et peuvent s'appuyer sur les informations et les services fournis dans la mesure spécifiée.

Le responsable d'AAS doit également s'assurer que toutes les informations et tous les services fournis par l'intermédiaire d'une Enveloppe de Gestion d'Actif sont conformes aux exigences de sécurité de l'actif.

6.3.3 Exigences relatives à la série IEC62443

6.3.3.1 Exigence

Une Enveloppe de Gestion d'Actif doit prendre en charge la mise en œuvre de la sécurité des IACS conformément à la série IEC 62443.

6.3.3.2 Justification et recommandations

La série IEC 62443 est la série prépondérante utilisée pour mettre en œuvre la sécurité des systèmes d'automatisation et de commande industriels (IACS). Pour les contrôles de sécurité non directement traités par la série IEC 62443, la série IEC 62443 fournit des références à des normes de sécurité supplémentaires telles que la série ISO/IEC 27000. La série ISO/IEC 27000 définit la mise en œuvre d'un système de management de la sécurité de l'information (ISMS).

La structure de l'Enveloppe de Gestion d'Actif doit en tenir compte et elle doit être préparée pour mettre en œuvre les exigences résultant de la série IEC 62443 et de la série ISO/IEC 27000.

L'Annexe C fournit des éléments de base de la série IEC 62443 et leur relation avec l'Enveloppe de Gestion d'Actif (voir l'Article C.1). La série IEC 62443 présente les exigences fondamentales (voir l'Article C.2) et les niveaux de sécurité (voir l'Article C.3). L'Enveloppe de Gestion d'Actif doit refléter à la fois les exigences fondamentales concernées et le niveau de sécurité qui s'applique aux informations et aux services fournis par l'Enveloppe de Gestion d'Actif (comparer à C.4).

Annexe A (informative)

Normes correspondantes

A.1 Sources possibles pour les Submodels et les modèles Submodel

A.1.1 Généralités

L'Article A.1 détermine les ensembles de SubmodelElements pertinents en fonction du cas d'utilisation. La source de la définition du concept de SubmodelElements est nécessaire pour comprendre la sémantique du Submodel et de ses SubmodelElements. De plus, il est important de tenir compte en général des liens réciproques entre SubmodelElements.

A.1.2 Différents ensembles de définitions de concept pour les SubmodelElements

A.1.2.1 Généralités

Une Enveloppe de Gestion d'Actif est capable de conserver des informations et des services pour différentes disciplines techniques, différentes phases du cycle de vie et différents scénarios d'application et d'analyse. Un grand nombre et une grande variété de concepts et de classifications des produits sont applicables.

Les répertoires de concepts sont la source partagée pour la sémantique des SubmodelElements et spécifient les concepts, les types de propriétés, les cardinalités, les relations entre les concepts et les classifications de produits. Cela est le cas dans les quatre catégories de publications suivantes (voir la Figure 7):

- normes;
- spécifications d'un consortium;
- spécifications d'un fabricant;
- spécifications d'une communauté ouverte.

Ces catégories sont caractérisées ci-dessous.

A.1.2.2 Normes

Les normes sont élaborées et publiées par des organisations élaborant des normes (OEN) nationales ou internationales, comme l'ISO, l'IEC, etc. La participation à l'élaboration des normes est ouverte à toutes les parties prenantes. Toute personne ou organisation peut accéder aux normes publiées.

Le délai d'exécution type pour la création ou la modification d'une norme est de trois ans, mais des procédures accélérées sont disponibles.

Les dictionnaires de concepts normalisés spécifiant les types de propriétés sont publiés dans différents domaines sous la responsabilité des comités techniques pertinents de ces OEN. L'IEC fournit la base de données pour la publication de structures de données sémantiques (Dictionnaire de données communes de l'IEC [IEC CDD]). Se référer à <https://cdd.iec.ch>.

La qualité du produit est garantie par la participation d'experts du monde entier et par un processus clairement défini impliquant le choix des parties prenantes des organismes nationaux participants.

L'IEC CDD est une norme en format de base de données gérée par une procédure de demande de modification (basée sur les Directives ISO/IEC, Partie 1 — Supplément IEC, Annexe SL), qui réduit la durée de la procédure d'importation et de conservation du contenu (classes et types de propriété) à 6 mois.

A.1.2.3 Spécifications de consortium

Les spécifications de consortium sont développées et publiées par des associations de parties prenantes. La participation à un consortium et l'élaboration de spécifications de consortium exigent, dans la plupart des cas, une adhésion. L'accès aux publications est réservé aux membres bien que certains documents puissent être accessibles librement.

Pour les classes et les types de propriétés, les organisations correspondantes sont appelées "organisation de classification de produits", comme ECLASS, ETIM, ECCMA, idea, dictionnaire JEITA ECALS.

Le délai d'exécution pour la création ou la modification d'une spécification de consortium dépend du consortium et des procédures appliquées. En raison d'un nombre limité de parties prenantes, le consensus peut être plus rapide que pour la normalisation internationale.

La couverture des domaines publiés dépend du consortium.

A.1.2.4 Spécifications du fabricant

Les fabricants possèdent de nombreux types de propriétés privées; d'autres peuvent être créés rapidement et peuvent également concerner une description d'actif spécifique.

Les classes et types de propriétés privées pour des actifs spécifiques qui ne peuvent pas être développés conjointement avec d'autres parties prenantes peuvent être définis comme faisant partie d'un dictionnaire du fabricant et exposés par certaines organisations de classification de produits telles que l'ECCMA.

A.1.2.5 Spécifications de communauté ouverte

Dans une communauté ouverte, une entité ou une personne peut participer et contribuer à l'effort de collaboration sans être membre d'un certain consortium ou organisation. Le travail résultant est mis à disposition sous une licence libre (par exemple Creative Commons). De plus, les données et le processus sont ouverts au public dans toutes les phases.

Les spécifications de communauté ouverte sont développées dans le cadre d'un processus similaire au développement de logiciels en licence libre. Ces spécifications de communauté ouverte définissent des concepts qui peuvent être identifiés par un URI, par exemple "http://admin-shell.io/className/version/revision/propertyName".

EXEMPLE Un exemple de spécification de communauté ouverte est
<https://github.com/admin-shell-io/id/tree/master/vdi/2770/1/0>.

A.1.3 Spécifications internationales existantes pouvant être utilisées comme base pour les modèles Submodel

Les Normes internationales spécifient les classes, les fonctions et les informations qui peuvent être représentées dans les SubmodelElements et sont donc importantes pour le concept d'Enveloppe de Gestion d'Actif (voir aussi la Référence [1], ISO/IEC TR 63306-1 et -2, ISO TR 23087:2018). Chacune de ces sources de SubmodelElements appartient à un certain domaine technique qui peut constituer un modèle Submodel et, par conséquent, un contexte pouvant être référencé. Cependant, les normes et les modèles Submodel peuvent être gérés de manière indépendante.

Du Tableau A.1 au Tableau A.3, voici quelques exemples:

- le Tableau A.1 énumère les normes fournissant directement des définitions des entrées de répertoire de concepts, par exemple au moyen de l'IEC CDD;
- le Tableau A.2 énumère les normes fournissant des sources potentielles d'entrées de répertoire de concepts, qui peuvent être référencées par des modèles Submodel. Ces sources potentielles exigent un certain approfondissement avant que des définitions spécifiques des entrées de répertoire de concepts ne puissent être dérivées;
- le Tableau A.3 énumère les normes fournissant des modèles de référence potentiels, qui ne définissent pas uniquement des entrées de répertoire de concepts, mais aussi des concepts et définitions généraux importants pour l'interopérabilité syntaxique.

Tableau A.1 – Exemples de normes fournissant des entrées de répertoire de concepts qui peuvent être référencées par des modèles Submodel

Spécification/norme		Domaine technique
Numéro	Titre	
IEC 62683 (toutes les parties)	Appareillage à basse tension – Données et propriétés de produits pour l'échange d'informations	Appareillage à basse tension
IEC 61987 (toutes les parties)	Mesure et commande des processus industriels – Structures de données et éléments dans les catalogues d'équipement de processus	Dispositifs de terrain pour la commande de processus
ISO 13399 (toutes les parties)	Représentation et échange des données relatives aux outils coupants	Outils coupants
ISO 13584-42	Systèmes d'automatisation industrielle et intégration — Bibliothèque de composants – Partie 42: Méthodologie de description: Méthodologie appliquée à la structuration des familles de pièces	Familles de pièces
ISO 13584-501	Systèmes d'automatisation industrielle et intégration – Bibliothèque de composants – Partie 501: Dictionnaire de référence pour les instruments de mesure – Procédure d'enregistrement	Instruments de mesure
ISO 13584-511	Systèmes d'automatisation industrielle et intégration – Bibliothèque de composants – Partie 511: Systèmes mécaniques et composants à usage général — Dictionnaire de référence pour les fixations	Systèmes mécaniques et composants
ISO/TS 15926-4	Systèmes d'automatisation industrielle et intégration – Intégration de données de cycle de vie pour les industries de "process", y compris les usines de production de pétrole et de gaz – Partie 4: Données de référence initiales	Industries de "process", y compris les usines de production de pétrole et de gaz
ISO 18582 (toutes les parties)	Transmissions hydrauliques et pneumatiques — Spécification d'un dictionnaire de référence	Composants de transmissions hydrauliques et pneumatiques
ISO 23584 (toutes les parties)	Optique et photonique — Spécification d'un dictionnaire de référence	Optique et photonique
ISO 25297 (toutes les parties)	Optique et photonique — Transfert électronique des données optiques	Optique et photonique