Report

Information model for functional safety

An APOS project report

Author(s):

Stein Hauge (SINTEF), Mary Ann Lundteigen (NTNU), Maria Ottermo (SINTEF), Shenae Lee (SINTEF), Stig Petersen (SINTEF)

Report No:

2023:00109

Client(s):

Multiclient

SINTEF Digital
Postal address:
Postboks 4760 Torgarden
7465 Trondheim, Norway

Switchboard: +47 40005100

info@sintef.no

Enterprise /VAT No:
NO 919 303 808 MVA

Report

Information model for functional safety

KEYWORDS
Information model
Digital Twin
Interoperability
Safety Instrumented Function
Asset Administration Shell

VERSION 1	**DATE** 2023-03-21

AUTHOR(S)
Stein Hauge (SINTEF), Mary Ann Lundteigen (NTNU), Maria Ottermo (SINTEF), Shenae Lee (SINTEF), Stig Petersen (SINTEF)

CLIENT(S) Multiclient	**CLIENT'S REFERENCE** Erik Korssjøen
PROJECT NO. 102020273	**NO. OF PAGES/APPENDICES** 48

SUMMARY
A main objective of this report has been to contribute toward further digitalisation and standardisation of the petroleum industry. In particular, the report explores:

- How ideas from Industry 4.0 and the AAS (asset administration shell) framework can be applied within the context of IEC 61511 and functional safety for the process industry
- Which data element to include in an information model for functional safety and how such a model can be structured and maintained throughout its lifecycle.
- How an information model for functional safety relates to and interacts with more overall information models also involving other users and disciplines.

PREPARED BY Stein Hauge, Mary A. Lundteigen, Maria V. Ottermo	SIGNATURE Stein Hauge
CHECKED BY Lars Bodsberg	SIGNATURE Lars Bodsberg
APPROVED BY Anita Øren	SIGNATURE Anita Øren

REPORT NO.	**ISBN**	**CLASSIFICATION**	**CLASSIFICATION THIS PAGE**
2023:00109	978-82-14-07941-8	Unrestricted	Unrestricted

Document history

VERSION	DATE	VERSION DESCRIPTION
Draft 01	2022-11-03	Draft for APOS and PSD member comments
01	2023-03-21	First official (open) version

Table of contents

Klikk eller trykk her for å skrive inn tekst.

Preface

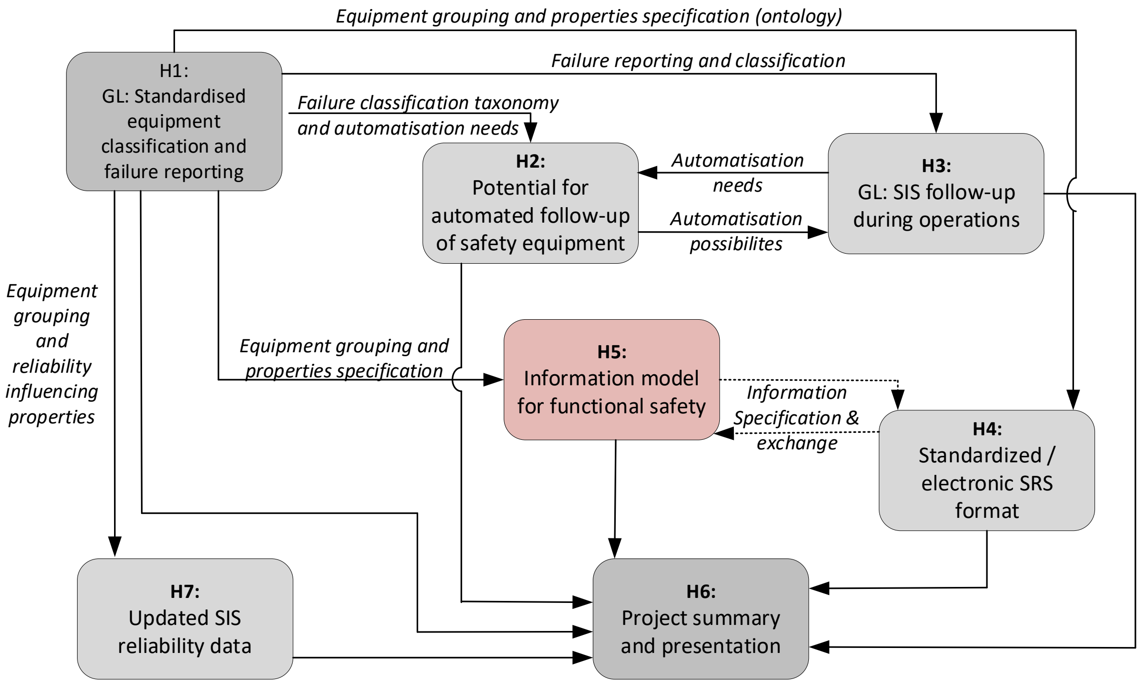
The work described in the report has been carried out as part of the research project “Automated process for follow-up of safety instrumented systems” (APOS). We would like to thank everyone for comments and valuable input to this work. The APOS project has received funding from the PETROMAKS 2 programme, The Research Council of Norway and PDS-forum.[1]

This report documents activity five (H5) in the APOS research project "Automated process for follow-up of safety instrumented systems (SIS)" (*Norw: Automatisert prosess for oppfølging av instrumenterte sikkerhetssystemer*). A main purpose of this project has been to simplify and standardise reporting and classification of SIS failures, including the classification of safety equipment, and to provide a basis for increased automation and standardisation of SIS follow-up, including a specification for an information model for functional safety. The APOS project comprises seven related activities:

1. H1: Guidelines for standardised equipment classification and failure reporting /14/
2. H2: Potential for automated follow-up of safety equipment /52/
3. H3: Guideline for follow-up of Safety Instrumented Systems (SIS) in the operating phase /51/
4. H4: Standardised/electronic SRS format /54/
5. **H5: Information model for functional safety (this report)**
6. H6: Project summary and presentation
7. H7: PDS Data handbook, 2021 Edition /53/

An overview of the relationship between the APOS activities is shown below.

[1] For more information about PDS-forum and the APOS project, reference is made to: https://pds-forum.com/


Equipment grouping and properties specification (ontology)
Failure reporting and classification
H1:
GL: Standardised equipment classification and failure reporting
Failure classification taxonomy and automatisation needs
H2:
Potential for automated follow-up of safety equipment
Automatisation needs
Automatisation possibilites
H3:
GL: SIS follow-up during operations
Equipment grouping and reliability influencing properties
Equipment grouping and properties specification
H5:
Information model for functional safety
Information Specification & exchange
H4:
Standardized / electronic SRS format
H7:
Updated SIS reliability data
H6:
Project summary and presentation

This report documents project activity five (H5).

Trondheim, March 2023

1 Introduction

1.1 Background and objective

A main objective of the H5 activity is to contribute toward further digitalisation of the petroleum industry, focusing on information modelling for the functional safety domain. As part of this, it is important for the industry to have a common understanding of what should be included in an information model for functional safety and how such a model can be structured and maintained throughout its lifecycle. It is also important to discuss how a functional safety information model, e.g., for a typical safety instrumented function (SIF), interacts and can be integrated with other disciplines' information models.

1.2 Abbreviations and terminology

Below, abbreviations applied in this report are explained.

AAS	Asset Administration Shell
AASX	File format used in the software AASx Package Explorer
AML	Automation Markup Language
APOS	*Norw*: Automatisert prosess for oppfølging av instrumenterte sikkerhetssystemer (Automated process for follow-up of safety instrumented systems)
API	Application Programming Interface
AutomationML	Automation Markup Language
CAEX	Computer Aided Engineering Exchange
CDD	Common Data Dictionaries
CMMS	Computerized Maintenance Management System
CDV	Committee Draft for Vote
COLLADA	Collaborative Design Activity
DC	Diagnostic coverage
DLOP	Device List of Properties
ECLASS	Classification system for products and services
EDDL	Electronic Device Description Language
FDI	Field Device Integration
FEED	Front End Engineering Design
HART	Highway Addressable Remote Transducer
HFT	Hardware Fault Tolerance
HTTP	Hypertext Transfer Protocol
I4.0	Industry 4.0
IEC	International Electrotechnical Commission
IDTA	Industrial Digital Twin Association
IM	Information Model
IMS	Information Management System
IRDI	International Registration Data Identifier
JSON	JavaScript Object Notation
LOP	List of Properties
MQTT	Message Queuing Telemetry Transport
NAMUR	German: Interessengemeinschaft Automatisierungstechnik der Prozessindustrie e.V.

	(international user association of automation technology and digitalization process industries)
NE	NAMUR Recommendation
OLOP	Operating List of Properties
OPC	Open Platform Communications
OPCF	OPC Foundation
OPC UA	Open Platform Communications United Architecture
OT	Operational technology
O&M	Operation and Maintenance
PA-DIM	Process Automation Device Information Model
PFD	Probability of Failure on Demand
PI	PROFIBUS & PROFINET International
PSD	Process shutdown system
RAMI	Reference Architectural Model for Industry
RIP	Reliability Influencing property
SAS	Safety and Automation System
SIF	Safety Instrumented Function
SIL	Safety Integrity Level
SIS	Safety Instrumented Systems
SLOP	Safety List of Properties
SRS	Safety Requirement Specification
UML	Unified Modelling Language
WG	Working Group
XML	Extensible Markup Language

Table 1 explains some of the terminology used in this report.

Table 1 Terms and definitions.

Term	Definition
Asset	Physical or logical object owned by or under the custodial duties of an organization, having either a perceived or actual value to the organization, /3/ Note: In the case of industrial automation and control systems, the physical assets that have the largest directly measurable value can be the equipment under control. IEC 63278-1 CDV (Committee Draft for Voting) /4/: Physical, digital, or intangible entity that has a value to an individual or an organization
Asset administration shell (AAS)	Standardized digital representation of the asset, corner stone of the interoperability between the applications managing the manufacturing systems. It identifies the Administration Shell and the assets represented by it, holds digital models of various aspects (submodels) and describes technical functionality exposed by the Administration Shell or respective assets /3/. Note: Asset Administration Shell and Administration Shell are used synonymously, and in this report, the short form admin shell is also commonly used IEC 63278-1 CDV /4/: Standardized digital representation of the asset

Term	Definition
CDD	IEC Common Data Dictionary is a metadata registry providing product classification and formalized product descriptions that can be used in the context of smart manufacturing and Industry 4.0 (Definition from Wikipedia).
Digital twin	A real-time virtual representation of a real-world physical system or process (a physical twin) that serves as the indistinguishable digital counterpart of it for practical purposes, such as system simulation, integration, testing, monitoring, and maintenance. (Definition from Wikipedia)
ECLASS	ECLASS is a cross-industry master-data business standard for products and services information to be exchanged in a computer-sensible form across all borders – across sectors, countries, languages and organizations. The ECLASS data dictionary is based on ISO 13584-42 and IEC 61360-2. It is used for the exchange of product data for procurement, eCommerce, engineering tools, etc. https://eclass.eu/support/technical-specification/data-model/conceptual-data-model
Information model	A generic definition or representation of the data and services for a particular type of physical asset, e.g., a process transmitter. It provides an abstract but formal representation of such assets including their properties, relationships and the operations that can be performed on them. In OPC UA (see below) an information model consists of nodes (see below) that define the information and services for an application. Note that an information model provides formalism to the description of a problem domain *without constraining* how that description is mapped to an actual implementation in software. There may be many mappings of the information model. Such mappings are called data models, and one example is object models (see below). https://en.wikipedia.org/wiki/Information_model
Instance (AAS context)	Concrete, clearly identifiable component of a certain type, /3/ Note 1: It becomes an individual entity of a type, for example, a device, by defining specific property values. Note 2: In an object-oriented view, an instance denotes an object of a class (of a type). IEC 63278-1 CDV /4/: Specific asset that is uniquely identified
Object model	A logical object-oriented interface to some service or system.
OPC	Open Platform Communications; interoperability standard for the secure and reliable exchange of data in the industrial automation space and in other industries. It is platform independent and ensures the seamless flow of information among devices from multiple vendors https://opcfoundation.org/about/what-is-opc/
OPC UA	OPC Unified Architecture; a platform independent service-oriented architecture [communication protocol] that integrates all the functionality of the individual OPC Classic specifications into one extensible framework. https://opcfoundation.org/about/opc-technologies/opc-ua/
OPC UA client	Consumes data and executes commands exposed by OPC UA servers
OPC UA node types	Nodes are the building blocks that OPC UA applies for defining an information model (IM). OPC UA has six different node types;

Term	Definition
	- *Objects*; A physical entity / application / collection of entities, such as a valve, an actuator, a pump, motor and pump, a SIF (input and logic and output), etc. - *Variables*; Defines a property of the object, e.g. dimension, material, setpoint or measuring principle. - *Data types*; Defines the property of the variable, e.g. boolean, integer, floating point, strings/text and composites. - *Methods*; Defines commands that are exposed to clients, e.g. start, run, hold, reset, etc. - *Events*; Defines events that OPC UA clients can access such as alarms and pre-alarms, errors/faults, statuses, etc. - *References*; Defines the relationships between nodes, e.g. a pump has a motor, a valve has an actuator, the transmitter has a property, the detector has a serial number, etc.
OPC UA server	An application that exposes data and services to OPC UA clients in a client/server architecture
OT (Operational Technology)	Technology that supports, controls and monitors industrial production, control and safety functions /43/.
Property (AAS context)	Defined characteristics suitable for the description and differentiation of products or components, /3/ Note 1: The concept of type and instance applies to properties. Note 2: This definition applies to properties such as described in IEC 61360 /46/ and ISO 13584-42 /47/ Note 3: The property types are defined in dictionaries (like IEC component Data dictionary or ECLASS /37/), they do not have a value. The property type is also called data element type in some standards. Note 4: The property instances have values, and they are provided by the manufacturers. A property instance is also called property-value pair by certain standards. Note 5: Properties include nominal value, actual value, runtime variables, measurement values, etc. Note 6: A property describes one characteristic of a given object. Note 7: A property can have attributes such as code, version, and revision. Note 8: The specification of a property can include predefined choices of values.
Submodel (AAS context)	Models which are technically separated from each other, and which are included in the asset administration shell, /3/ Note 1: Each submodel refers to a well-defined domain or subject matter. Submodels can become standardized and thus become submodel templates. Note 2: Submodels can have different lifecycles. Note 3: The concept of template and instance applies to submodels. IEC 63278-1 CDV inspired /4/: A representation of an aspect of an asset used to organize the information and services within an AAS into distinct parts.
Type (AAS context)	Hardware or software element which specifies the common attributes shared by all instances of the type, /3/
View (AAS context)	Projection of a model or models, which is seen from a given perspective or vantage point and omits entities that are not relevant to this perspective, /3/.

1.3 Content of this report

The content of this report is briefly described below:

- Chapter 2, "Information models, data models and digital twins" attempts to clarify some relevant terms and expressions related to information modelling and digitalisation.
- Chapter **Error! Reference source not found.**, "Information model - SIF case" describes the possible content of a SIF information model in the context of IEC 61511 and follow-up of SIF/SIS from design throughout operation.
- Chapter 4, " Functional safety information model versus other disciplines" briefly discusses the interfaces and possible integration between an information model for functional safety assets and corresponding information from other disciplines.

2 Information models, data models and digital twins

For people working within automation and functional safety, it may feel frustrating and somewhat overwhelming when taking on new topics like digitalization and information modelling. Terms and expressions are being used interchangeably and the meaning and differences are often unclear. In this chapter an attempt has therefore been made to provide some clarifications.

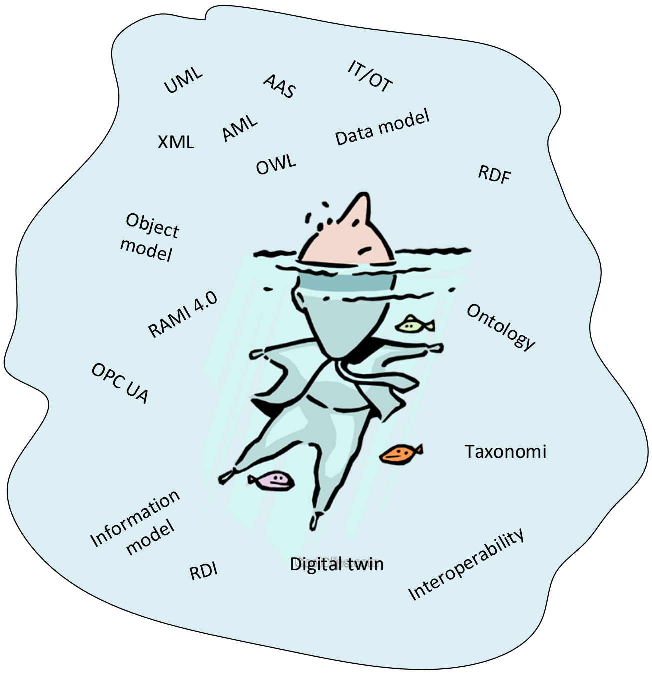


Figure 1 **An ocean of terms, expressions, and abbreviations**

2.1 Information model versus data model

An *information model* is used to model individual assets (or objects), such as components, systems, facilities, buildings, process plants, etc. at a conceptual level. For the asset under consideration such a model provides an abstract representation of relevant asset data/information that can include properties, relationships, relevant constraints and rules, and the operations that can be performed on them. The content and degree of detail of the abstractions presented in the information model will largely depend on the needs of the users of the model.

An information model provides formalism to the description of some asset *without* constraining how that description is mapped to an actual implementation in software (e.g. an information model is independent of any specific protocols used to transport the data). There may be several mappings of the information model. Such mappings are *called data models*, /30/.

Data models are normally defined at a lower level of abstraction and will include more details as compared to information models. Data models are mainly intended for implementors and include implementation- and protocol-specific constructs. Such models are often represented in formal data definition languages that are specific to the data protocol being used.

Although information models and data models serve different purposes, it is not always easy to decide which detail belongs to an information model and which belongs to a data model. Similarly, it is sometimes difficult to determine whether an abstraction belongs to an information model or a data model, /30/.

2.2 Industry 4.0

Germany has carried out intensive development work for many years and is a world leader in integrating individual systems solutions through its Platform Industry 4.0 initiative. The vision of Platform Industry 4.0 encompasses a massive digitalization process described through the three dimensions of the Reference Architecture Model for Industry 4.0 (RAMI 4.0). RAMI (Reference Architectural Model for Industry) 4.0 describes how an asset can be converted into a digital representation and processed in the digital world by structuring the assets along the three main axes shown in Figure 2:

- Layers: Representation of the different virtual (or digital) mappings associated with the asset.
- Lifecycle & value stream: Representation of the asset through the lifecycle – from type definitions to instances
- Hierarchy levels: Representation of the main functional levels, organized similarly as the network levels of the Perdue Reference Architecture

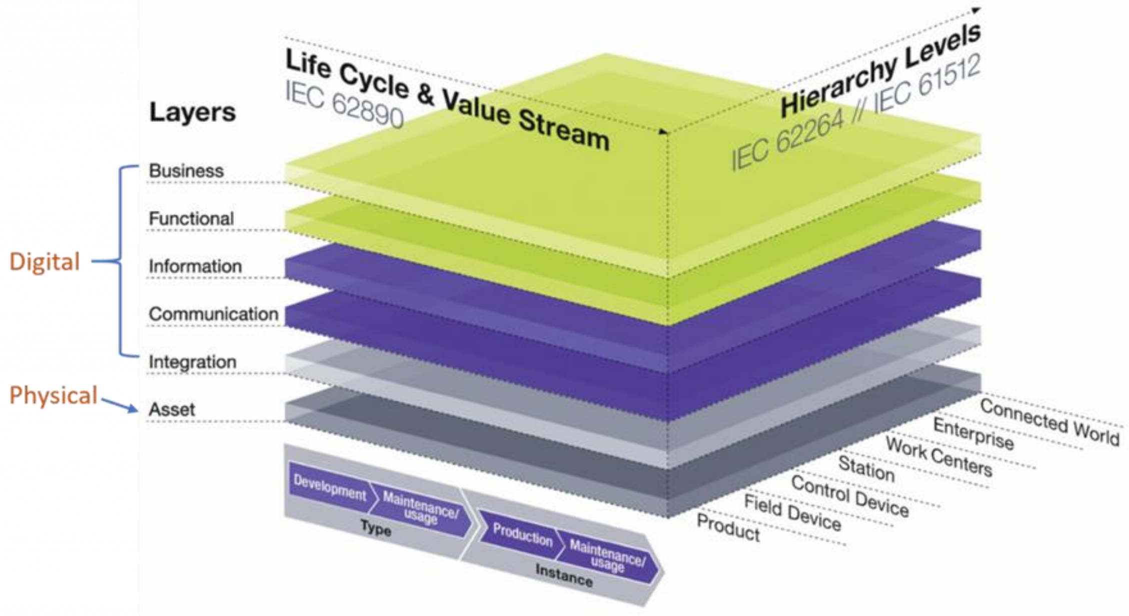


Figure 2 **3D representation of the Reference Architecture Model Industry 4.0 (RAMI4.0) /34/**

Every asset within a plant from the physical world is represented in the information world by a digital and uniquely identifiable counterpart – a digital twin. In Industry 4.0 this digital representation of the asset is known as the Asset Administration Shell (AAS). The AAS provides a unique identifier for this asset and a general interface to access the information and functionalities of the asset.

The German Industry 4.0 (I4.0) initiative has so far focused on the manufacturing industry. In this report we consider information modelling for the process industry, related to functional safety, and discuss the potential use of the I4.0/AAS framework within this context.

2.3 Digital twins

The term “digital twin” was first introduced with product lifecycle management in 2003 (Grieves (2015), /19/). Since then, the industry has embraced the concept as part of the digitalization effort of products, systems, and plants. On Wikipedia, a digital twin is described as "a real-time virtual representation of a real-world physical system or process (a physical twin) that serves as the indistinguishable digital counterpart of it for practical purposes, such as system simulation, integration, testing, monitoring, and maintenance." In practice, the literature reveals several interpretations of a digital twin depending on the context. Therefore, the refinement suggested by Kritzinger et al (2018), /20/, can be helpful: They suggest three categories of digital representation measured by the level of automatic interaction with the real system (or asset):

- *Digital model*, where the interaction between a digital representation and the real system is fully manual. This means that there is no *automatic* update of the digital model if the state of the real system changes and simulation results from the digital model is not used automatically as input to the real system
- *Digital shadows or mirrors*, where the interaction between the digital representation and the real system is partially automatic: The digital representation automatically receives information about the states of the real system, but the other way around is managed offline or manually.
- *Digital twin*, where the interaction between the digital representation and the real system is fully automatic: The digital representation automatically and online receives the status of the real system, and the real system automatically receives data and information that are relevant for optimization, control, and decision-making from analyses and simulations.

A visualization of the three categories is shown in Figure 2.

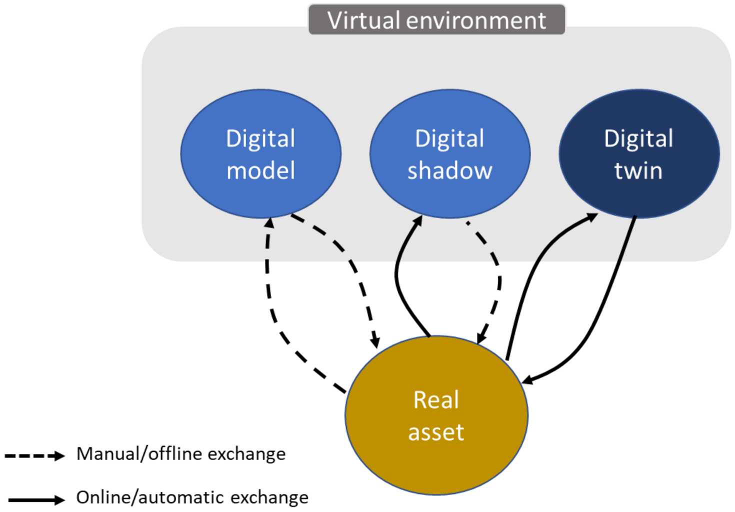


Figure 3 Digital representations dependent on the level of integration (Adapted from /9/)

The three categories can be seen as evolvements of a digital twin in a system's lifecycle: A single or a set of digital models is generated in the design phase by product developers and system integrators, while their digital shadows and digital twins connect the digital models to the real system throughout the manufacturing, installation, and operational phase. The aim of the digital twin is to manage all information related to one or more assets and provide active decision support to the real system such as e.g., predictions, scenario analysis, and optimisation.

Wright and Davidson (2022) /38/ also discuss the difference between a model and a digital twin. As part of this, they discuss some digital twin definitions which point to three important parts of a digital twin:

- a model of the physical object (or asset) under consideration,
- an evolving set of data relating to the object, and
- a means of dynamically updating or adjusting the model in accordance with the data.

In their discussion Wright and Davidson (2022) emphasis that a digital twin must have a physical counterpart and from this they conclude that *a digital twin without a physical twin is a model*. From this they infer that “digital twins for design” is only meaningful when the prototyping stage is reached.

Wright and Davidson (2022) further state that in general, the model for a digital twin should be:

- sufficiently physics-based such that updating parameters within the model based on measurement data is a meaningful thing to do,
- sufficiently accurate such that the updated parameter values will be useful for the application of interest, and
- sufficiently quick to run such that decisions about the application can be made within the required timescale.

For safety verifications or safety performance modelling, they point out that high accuracy will often be more important than a short run time because the safety-critical models are usually run less frequently. This coincides with e.g., verification/updating of SIL (Safety Integrity Level) requirements which is a slow process but depends on accurate input concerning failure history (and correct failure classification).

Based on the above, some preliminary conclusions can be made:

- A digital twin relies on the ability to integrate digital models from many providers and exchange data and information efficiently over the system lifecycle. A keyword is therefore interoperability, i.e. the ability of the interacting systems, software, and models to exchange and make use of information[2]. Probably the most promising standardisation initiative for creating digital representations for full interoperability is the Asset Administration shell (AAS), which is I4.0's digital representation of an asset as mentioned in the previous section. See Table 1 and the next section for further definitions and description of AAS.
- Wright and Davidson (2022) indicate that a digital twin must have some calculation/ simulation model inherent in it. This implies that in their view an AAS may often be more of a digital representation than a true digital twin. We will leave this - possibly academic - discussion for now but observe that there is some debate as to what is a true digital twin.
- Wright and Davidson's end conclusion however seems to be generally applicable "*Successful deployment of digital twins will require trust in the model, trust in the data, and trust in algorithms used to update the model based on the data*".

2.4 The asset administration shell (AAS)

The Asset administration shell (AAS) is often explained as a standardized framework for generating digital representations of assets. Assets represent something of value to an organization in the real (physical) world, ranging from physical devices and systems to software, documentation, and licenses. The digital representations can generate additional value from the assets by exploiting data and information for monitoring, analysis, and predictions.

[2] Adapted from Oxford English Languages

The AAS framework satisfies two basic requirements:

- *Being machine-readable:* The framework requires the use of standardized machine-readable formats.
- *Being interoperable:* The framework specifies data exchange formats that are suitable for use across the value chain and lifecycle phases, and between the plant floor and cloud applications.

The AAS framework is described thoroughly in several reports and whitepapers from Platform Industry 4.0 (https://www.plattform-i40.de/). Furthermore, an IEC standard IEC 63278 is under development and will eventually comprise three parts:

- IEC 63278-1: Asset Administration Shell for industrial applications – Part 1: Asset Administration Shell structure /4/.
- IEC 63278-2: Asset Administration Shell for Industrial Applications – Part 2: Information meta model /5/.
- IEC 63278-3: Asset Administration Shell for Industrial Applications – Part 3: Security provisions for Asset Administration Shells /6/.

Three types of AAS have been defined:

- Type 1: The AAS shell contains only static information, often received, and converted from formats such as AutomationML, JSON (JavaScript Object Notation) etc.
- Type 2: The AAS shell contains both static and dynamic information and is in a server architecture using technologies like OPC UA. It relies on vertical server-client or (more likely) a publisher-subscriber type of management.
- Type 3: The AAS shell contains both static and dynamic information (as for type 2). Unlike type 2, type 3 AAS manages the data exchange autonomously – both vertically and horizontally, /35/. This means that the AASs themselves initiate and agree with other AAS about when to exchange what type of data.

As far as we understand, type 3 is still for future implementation, while type 2 can be adapted with existing technologies.

Figure 4 visualizes some of the key features of the AAS:

- The digital representation of an asset, such as asset A and B, has a finite scope represented visually by a **shell-like** symbol.
- The shell organizes the asset characteristics like properties, parameters, and functions into a set of **submodels.**
- The shells with their submodels aggregate to an overarching **information model** that can be distributed across many hardware and software platforms.
- Unique interpretation of commonly used characteristics is made by referencing **standardized semantics** placed in a repository such as the common data dictionaries **(CDD)** /32/ based on IEC 61987 /45/ (list of properties)**.**
- Submodels are organized to satisfy needs of **use cases.** Use cases are example of services which can be provided with the data from the assets. For example, data collected about a valve can be used to monitor the valve's technical condition, the safety performance, and the operating costs of maintenance. Each of these services require some specific type of data not necessarily needed by other users.
- The interoperable data exchange format ensures data exchange between assets. The shells implement **AAS interfaces** for this purpose.

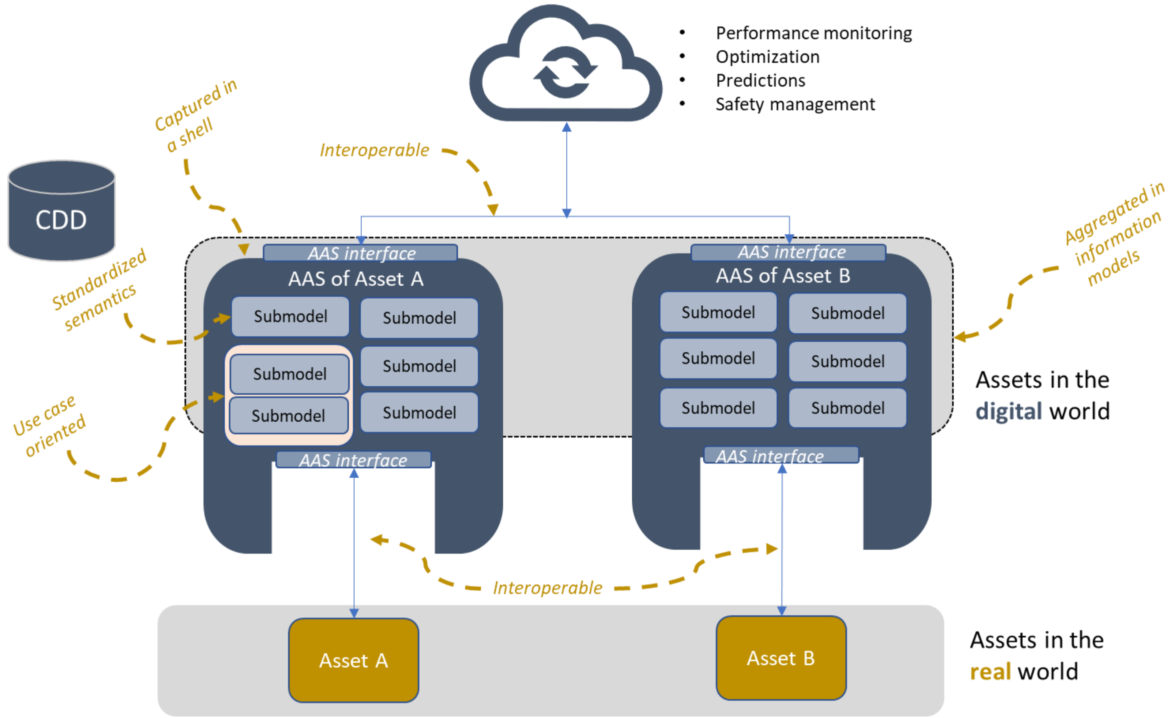


Figure 4 **Key attributes of** the **AAS**

The *AAS metamodel* is a cornerstone in the realization of the AAS. The metamodel specifies the formats and the modelling rules in UML (Unified Modelling Language) for construction of shells, submodels, and relationships to assets and CDD. The shells and submodels can be made as templates and then instantiated for specific assets. As templates, the shells and submodels may be stored in inhouse or open repositories. As instances they can be stored in one or more servers. The overview of relationships between assets, shells, and submodels are stored in repositories, either locally/inhouse or globally/open. The repositories provide the addresses to the server endpoints where the shells and submodels are located.

Software tools have been developed to generate AAS environments. For example, the AASx package explorer /42/ is an open-source software to set up and specify shells with static information. BaSyx /41/ is another open-source tool referred to as a middleware for putting the AAS in a live/run time environment. It connects shells and submodels with assets and applications using OPC UA, HTTP (Hypertext Transfer Protocol) or BaSyx native technologies.

The realization of AASs adopt open standards (including Industry 4.0), such as OPC UA, Automation ML /39/, JSON /40/, and MQTT (Message Queuing Telemetry Transport) /50/. **Figure 5** gives some examples of how these standards are used. For example, information like device technical specifications, drawings, and schematics can be transferred *into* the AAS format if modelled in Automation ML or JSON. Live data exchange across network layers and applications is often implemented with OPC UA, MQTT, or similar service-oriented communication. AAS shells and information models are then converted to e.g., OPC UA information models. The main advantage of modelling it all in an AAS environment first, is the ability to make specifications and formats independently of future realizations.

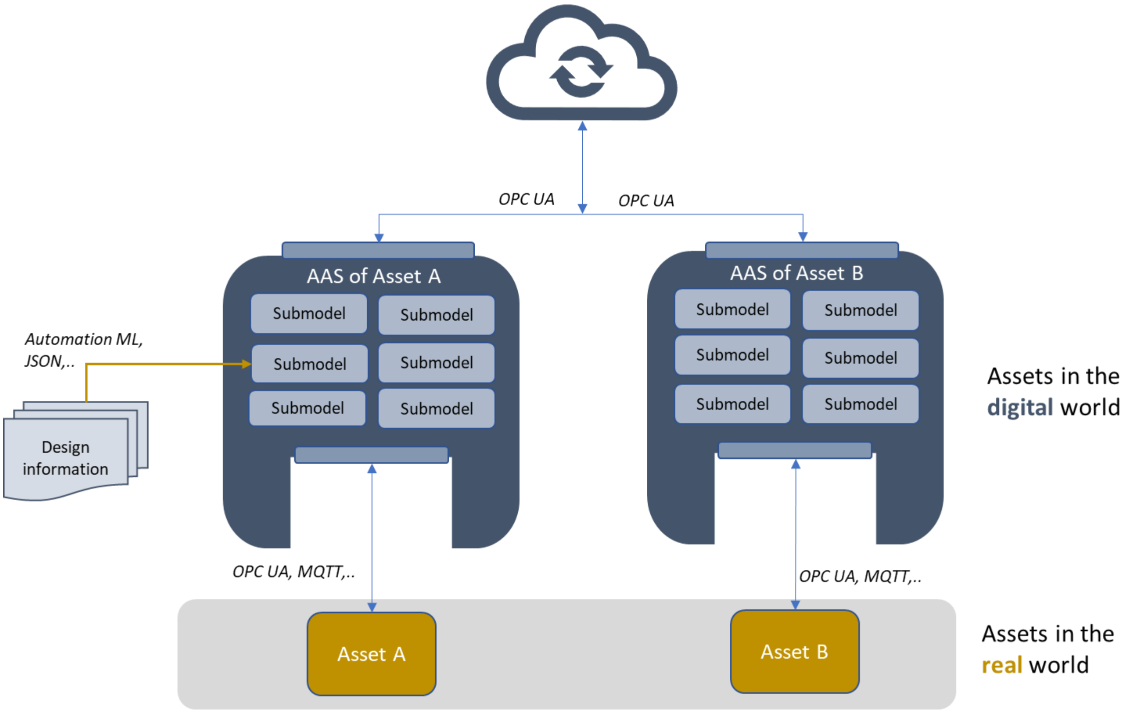


Figure 5 **Data exchange formats with AAS**

The scope for implementing AAS will potentially be extensive and involve several actors and developers. Therefore, it is important to evaluate the realism and challenges related to the development of AAS by use of relevant and limited use cases. It is also important to be involved in international standardization work (e.g., IDTA and NAMUR) and that the process is controlled to avoid that a lot of different actors (e.g. IT servicing companies) develop their own proprietary (non-standardised) AASs for functional safety etc.

The two next sections provide an overview of the AAS technology mapping of the two common industry standards OPC UA and AutomationML.

2.5 Open platform communications united architectures (OPC UA)

OPC (Open Platform Communications) in its original form was developed in the mid-1990s addressing the need for a standardized interface for the multitude of fieldbus specifications flooding the market at the time (e.g., 4-20mA, HART (Highway Addressable Remote Transducer), Foundation Fieldbus, Modbus, and PROFIBUS). The main task of OPC was to translate generic system commands to device specific commands, and vice versa. In 2008, the OPC Foundation released a new set of standards, OPC Unified Architecture (UA), following the industry trends of independency from vendors, products, and technologies. OPC UA is an open platform architecture, widely used for communication between OT (Operational technology) equipment and systems (not including field devices).

OPC UA consists of four models to facilitate information and data exchange /22/:

- An *information model* to represent data structures, behaviour, and semantics.
- A *message model* for interaction between applications.
- A *communication model* for data transfer between endpoints.
- A *conformance model* for interoperability between systems.

The communication setup in OPC UA can be either a traditional Client-Server configuration or a Publish-Subscribe configuration where the data source and potential data users are decoupled by a Message Oriented Middleware.

2.5.1 The OPC UA information model

OPC UA provides a framework where information can be represented as *Objects* in an *AddressSpace* accessible through standard OPC UA services. The OPC UA *Objects* consist of *Nodes* connected by *References*, and different types of Nodes are used to convey different semantics /23/:

- *Object node:* represents a system, a system component, as a real-world object, and even as a software object. Objects may contain variables, methods, and references to other related objects.
- *Variable Node*: represents a value that can be read or written, along with an associated *DataType.*
- *Method Node*: represents a function that can be called.

An example *Object* for a "Reservation" with associated *Method* and *Variable Nodes* is illustrated in Figure 6.

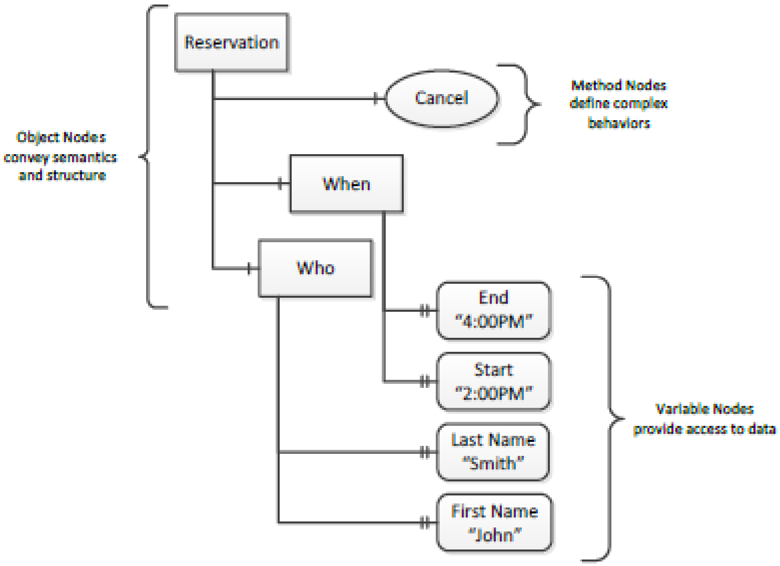

Reservation
Cancel
Method Nodes define complex behaviors
Object Nodes convey semantics and structure
When
Who
End "4:00PM"
Start "2:00PM"
Last Name "Smith"
First Name "John"
Variable Nodes provide access to data

Figure 6 A Basic Object in an OPC UA Address Space /24/

The OPC UA information model defines a wide range of functionality, and it is not expected that all *Clients* or *Servers* support all available functionality. However, OPC UA includes the concept of *Profiles* that define a minimum set of functionalities required for different applications or use. In addition to *Profiles*, OPC UA also has companion specifications for different industry verticals, describing an Information Model by defining relevant *ObjectTypes*, *VariableTypes*, and *DataTypes*.

2.5.2 Open platform communications platform united architecture (OPC UA) and AAS

To assist in the interaction and interoperability between OPC UA and AAS systems, the OPC Foundation released the companion specification "*OPC 30270 - OPC UA for Asset Administration Shell (AAS)*" in June 2021 /24/. The specification presents an extension to OPC UA, defining an OPC UA information model that conforms to the AAS metamodel.

The core part of the OPC 30270 is the description of the mapping of the AAS metamodel to the OPC UA information model for representation of AAS models and submodels within the address space of OPC UA servers. This is expressed in a set of rules for mapping AAS UML class diagrams to OPC UA information model elements, which must be followed in order to implement I4.0 conformant digital twins within the OPC UA framework.

The rules, 27 in total, are structured into seven categories according to which aspects of the AAS framework they cover /24/:

- General Rules
- Rules for *SubmodelElements*
- Rules for *Referables* and *Identifiables*
- Rules for *Qualifiables*
- Rules for *semanticId* and *Concept Descriptions*
- Rules for *Data Specifications*
- Rules for *Semantics of Metamodel Elements*

As a reference of the format and syntax of the rules, rule number 1 is defined as follows:

1. For all class elements of the metamodel, an *ObjectType* with the same name + suffix “Type” + prefix “AAS” is added. Example: AASAssetType for Asset, AASSubmodelElementType for SubmodelElement and AASQualifierType for Qualifier. These Types are derived from OPC UA’s BaseObjectType. Exception: ConceptDescriptions and Referables (see below).

Based on these rules, a general overview of AAS in the OPC UA information model has been established, as illustrated in Figure 7 (the AAS root), Figure 8 (AAS asset type description), Figure 9 (AAS submodel type), and Figure 10 (AAS submodel element types).

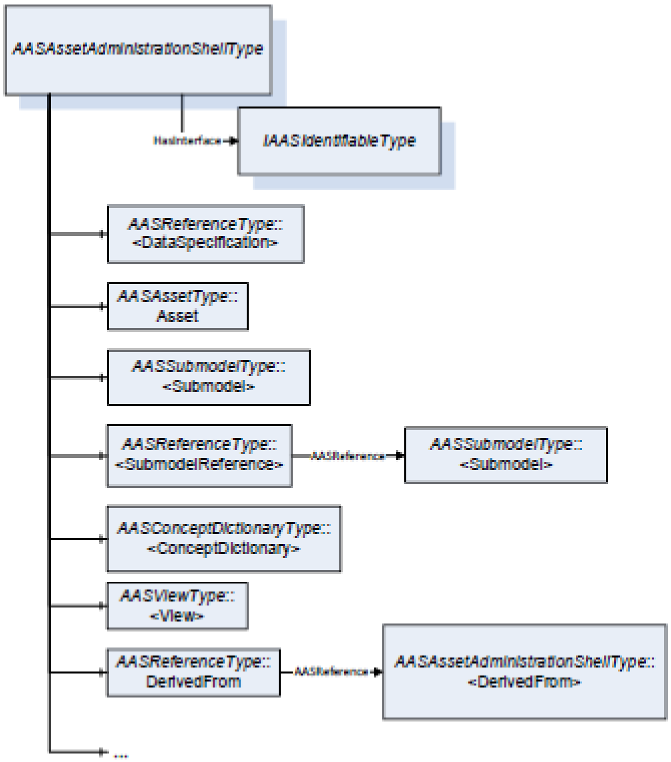


Figure 7 AAS root in the OPC UA information model /24/

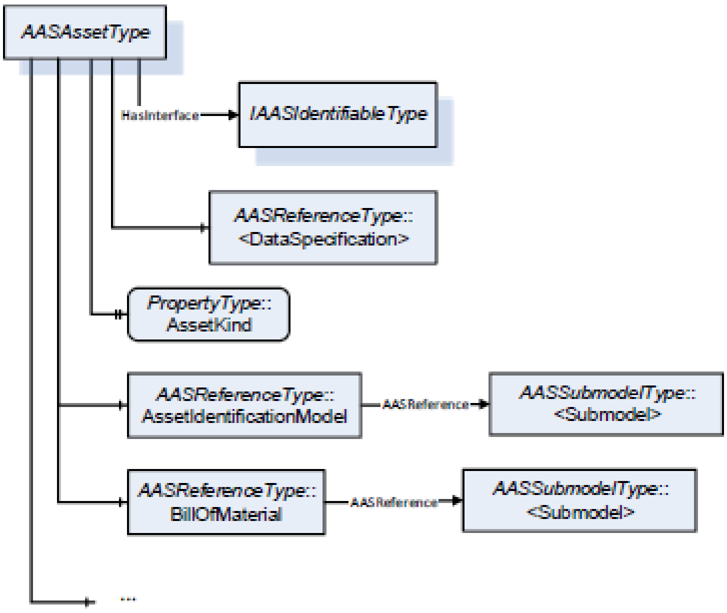


Figure 8 AAS asset type description in the OPC UA information model /24/

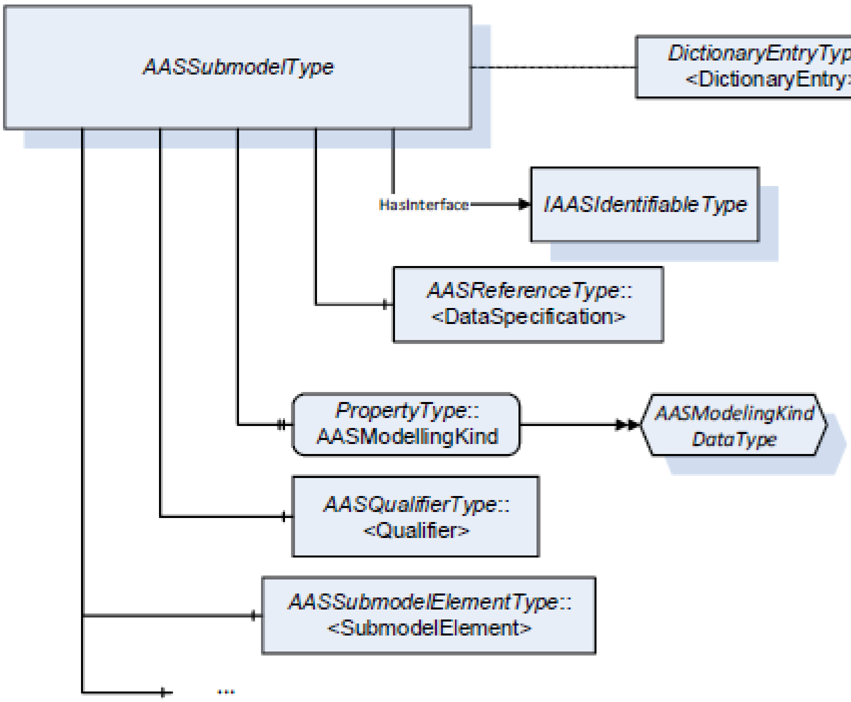


Figure 9 AAS submodel type in the OPC UA information model /24/

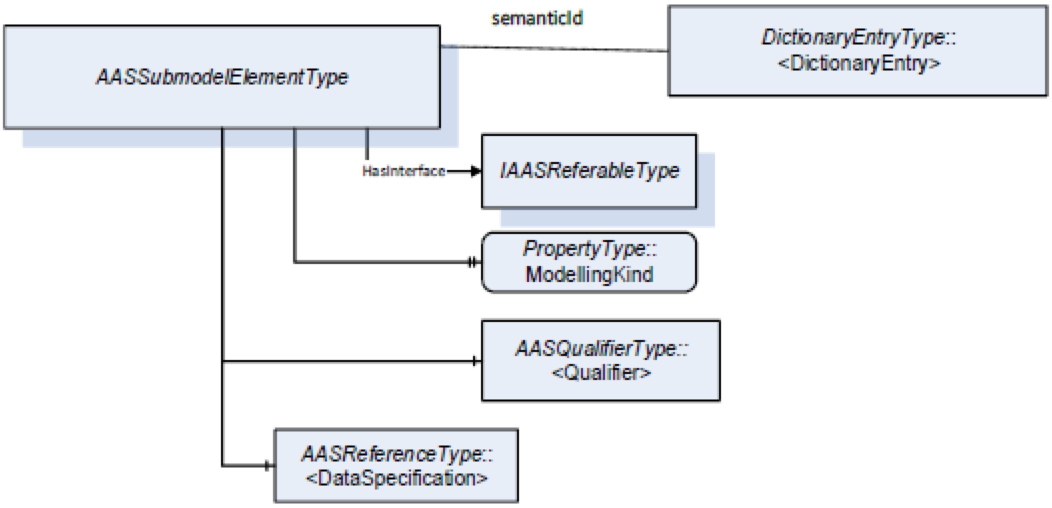


Figure 10 AAS submodel element types in the OPC UA information model /24/

As a remark, the AAS model elements representing security aspects are not part of the current version of the OPC 30270. It is expected that they will be included in a future revision.

Given the nature and format of the mapping rules for converting the AAS metamodel to the OPC UA information model, it should be possible to generate software parsing scripts for automating the procedure.

2.6 Automation markup language (AML)

Automation Markup Language (AutomationML, or AML), standardized in the IEC 62714 series /44/, is an XML (Extensible Markup Language) based data format developed for data exchange between different automation engineering tools. The main goal of AML has been to facilitate the interconnection of engineering tools from different disciplines, e.g., mechanical plant engineering, electrical design, process engineering, process control engineering, HMI development, PLC programming, and robot programming /25/.

The engineering information stored in AML follows an object-oriented approach, which enables modelling of real plant components as data objects. An AML object may consist of other sub-objects, and an object can be part of a larger composition or aggregation. The AML object can describe various physical entities, e.g., signal, PLC, tank, control valve, robot, manufacturing cell, or a complete site or plant. Typical plant automation objects include information on topology, geometry, kinematics, and logic. In turn, logic comprises sequencing, behaviour, and control.

The top-level data format of AML is CAEX (Computer Aided Engineering Exchange), standardized as IEC 62424 /26/. CAEX is a data format that allows storage of hierarchical object information, e.g., the interconnected modules and components of a plant. Like AML, CAEX is object-oriented with support for concepts such as classes, instances, encapsulation, and inheritance.

The geometry and kinematics information in AML is stored using the COLLADA (Collaborative Design Activity) file format, standardized as ISO 17506 /27/. COLLADA defines an XML-based schema for exchanging digital 3D assets between different graphics software applications.

The AML logic information describes sequences of actions and the behaviour of objects (e.g., I/O connections, and logical variables). The sequences are described and stored in external PLCopen XML documents, while variables and signals are published as CAEX external interfaces.

The basic architecture of AML, including the distribution of topology, geometry, kinematics, and logic information is illustrated in Figure 11.

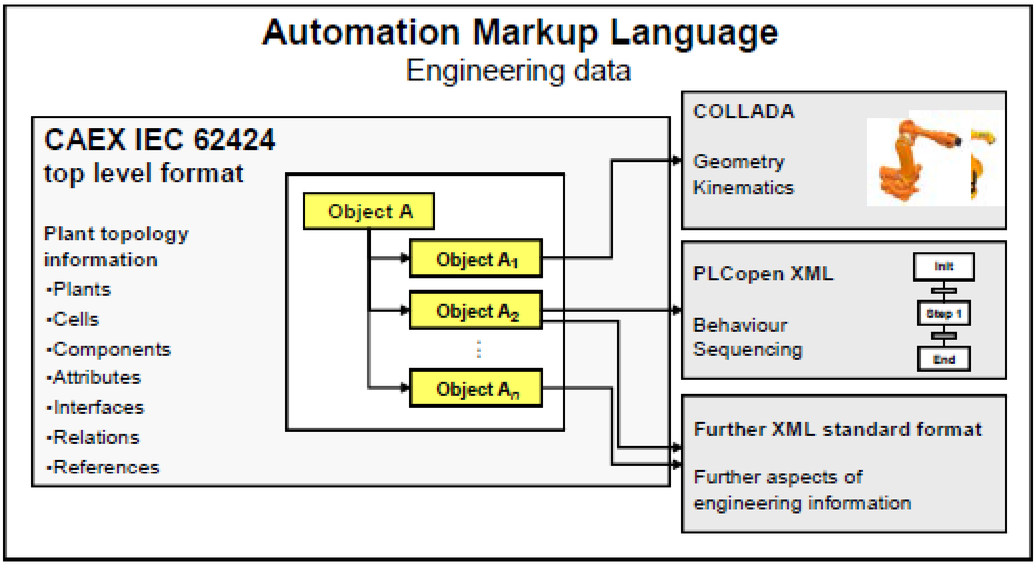


Figure 11 Overview of the AML engineering data exchange format /28/

2.6.1 Automation markup language (AML) and asset administration shell (AAS)

In November 2019, the AML consortium released the application recommendation "AAS Representation" describing how AML can be used for representing, storing, and exchanging AAS metamodel data /29/. Intended as a serialization format in the engineering phase, the document defines relevant mapping rules, and their related roles, interfaces, and system unit classes.

The rules for mapping the AAS information model to the AML information model are divided into seven categories according to which topics they cover /29/:

- General Rules for mapping
- Rules for element other than SubmodelElements of the AAS
- Rules for subtypes of AAS SubmodelElement
- Rules for the instance hierarchy of AML
- Rules for the Role Class Library of AML
- Rules for System Unit Class Libraries
- Rules for Interface Libraries

There is a total of 32 rules, and as a reference to the format and syntax of the rules, rule number 1 and 2 are defined as follows:

1. **If present, AML role class and attribute name are taken from the AAS metamodel.**
2. **If present, AML element names are the same as the value of idShort information from the AAS.** If not present, a sufficiently unique element name is to be generated.

Furthermore, the document defines and standardizes the AML libraries that must be used for the mapping with AAS. It is important to note, however, that security topics are not yet covered in the initial release of the technology mapping of AAS to AML. It is expected to be included in a future revision of the document.

Given the nature and format of the mapping rules for converting the AAS metamodel to the AML information model, it should be possible to write parsing software scripts for automating the procedure.

3 Functional safety information models

3.1 Functional safety assets in the context of IEC 61511

Like the generic IEC 61508 standard, /2/, IEC 61511, /1/ describes a set of safety lifecycle processes and activities to manage process risk. The safety lifecycle describes a risk-based approach to the identification, realization, and follow-up of SIFs and their performance requirements, with the phases and activities identified in the upper part of Figure 12 below (in green colour). One of the purposes of introducing AAS is to identify, structure, and manage data, information, models, and analyses *of assets* within and across these phases. This points to one of the first challenges of introducing AAS: What is to be regarded as assets?

As a starting point, Figure 12 proposes some potential functional safety assets and associated AASs, both for digital assets, such as documents, drawings, databases, certificates, and procedures (blue colour), and for physical assets (yellow colour administration shells). The bullet-listed items within the AASs are examples of what *may be* relevant submodels.

It should be noted that Figure 12 is *mainly for illustration*. The most practical way of structuring the AAS and the submodels must be further explored, also see discussion in section 3.3.

Concerning the lifecycle phases, the administration shells are in Figure 12 located where they typically first appear, but the AASs will be enriched with new information and use cases throughout the lifecycle of the asset, hence the (updated) AASs may exist across phases (exemplified in the figure by the functional safety management admin shell).

Different users will have different information needs and can therefore have different *views* into an AAS and the associated submodels. For example, a process transmitter can have hundreds of properties, of which only some 30 may have relevance for functional safety. The rest of the properties can be contained within the same transmitter admin shell and be accessed by other users and disciplines. The interfaces and integration between submodels, shells and properties relevant for functional safety and those of other disciplines are further discussed in chapter 4.

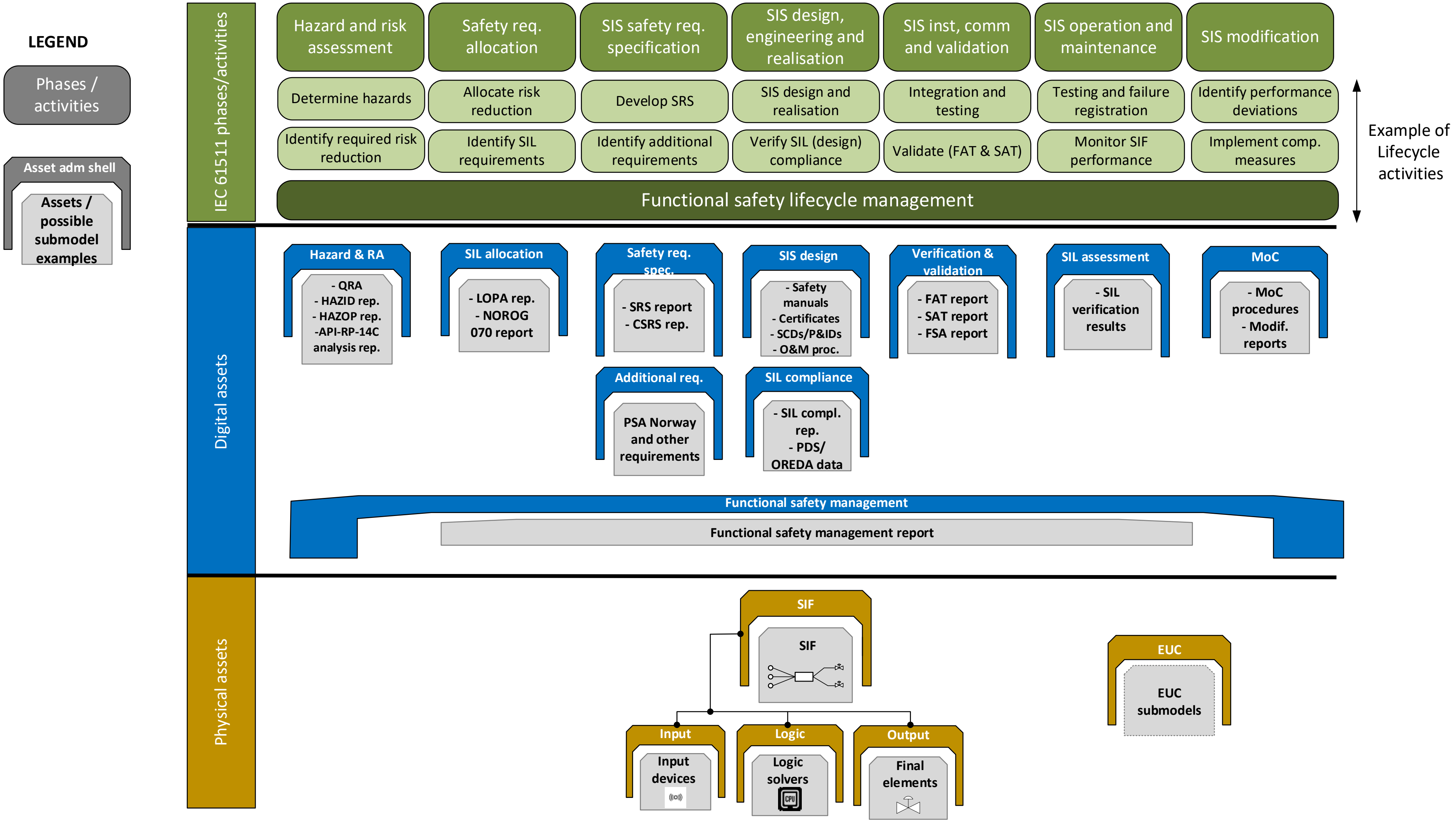

LEGEND
Phases / activities
Asset adm shell
Assets / possible submodel examples
IEC 61511 phases/activities
Hazard and risk assessment
Safety req. allocation
SIS safety req. specification
SIS design, engineering and realisation
SIS inst, comm and validation
SIS operation and maintenance
SIS modification
Determine hazards
Allocate risk reduction
Develop SRS
SIS design and realisation
Integration and testing
Testing and failure registration
Identify performance deviations
Identify required risk reduction
Identify SIL requirements
Identify additional requirements
Verify SIL (design) compliance
Validate (FAT & SAT)
Monitor SIF performance
Implement comp. measures
Functional safety lifecycle management
Example of Lifecycle activities
Digital assets
Hazard & RA
- QRA
- HAZID rep.
- HAZOP rep.
-API-RP-14C analysis rep.
SIL allocation
- LOPA rep.
- NOROG 070 report
Safety req. spec.
- SRS report
- CSRS rep.
SIS design
- Safety manuals
- Certificates
- SCDs/P&IDs
- O&M proc.
Verification & validation
- FAT report
- SAT report
- FSA report
SIL assessment
- SIL verification results
MoC
- MoC procedures
- Modif. reports
Additional req.
PSA Norway and other requirements
SIL compliance
- SIL compl. rep.
- PDS/ OREDA data
Functional safety management
Functional safety management report
Physical assets
SIF
SIF
EUC
EUC submodels
Input
Input devices
Logic
Logic solvers
CPU
Output
Final elements

Figure 12 Functional safety assets, submodels, and associated administration shells in the context of IEC 61511

3.2 Safety instrumented function (SIF) case

The information model discussed in this chapter focuses on the safety instrumented function (SIF) as a typical asset. This is based on work performed in the APOS project as well as results from an AAS pre-study (for Equinor) where the use of AAS for a generic safety instrumented function (SIF), with sensors, logic solver, and final elements was explored.

3.2.1 Use case and safety instrumented function (SIF) as a composite asset

As seen from Figure 12 several different AASs may be relevant throughout the functional safety lifecycle. IEC 63278-1 CDV /4/ argues that the identification and structuring of AAS, including submodels, should be based on the needs from use cases. Despite being frequently discussed in many AAS reports, the concept of *use case* remains a bit vague. With basis in /7/ (p.4-6) and /8/ (p. 8), we suggest the following interpretation:

- A use case is a specific function, as a service or operation, provided for a particular asset or a set of interacting assets.
- An asset can provide several services and operations, depending on the context or a given lifecycle phase. This implies that several use cases can be identified for the same asset.

AAS reports and standards indicate that submodels are the main means of constructing use cases of an asset. However, this does not imply that all submodels must be allocated to a single AAS. The allowance of a composite structure of AASs for a “systems-of-systems” type asset (e.g. a SIF composed of three types of assets: sensors, logic solver(s), and final element(s) indicates that use cases can be defined at different levels.

The selected *use case* in this chapter relates to the SIF asset. A SIF can, as already indicated, be considered a *composite asset* comprising the following assets: sensor(s), logic solver(s), and final element(s) as illustrated at the bottom of Figure 12. Hence there is a SIF level and an underlying equipment (or component) level. Note that aggregating all components of SIFs that belong to the same system (e.g. Process shutdown system (PSD)) creates another composite asset at the system level (e.g. "PSD system").

3.2.2 Safety instrumented function (SIF) information model

In the following, properties and parameters required to establish a "complete" information model for operational follow-up of a SIF, and how these properties and parameters could be categorized in a suitable manner, are discussed. For this purpose, two levels are discussed, the SIS equipment level and the (composite) SIF level.

Figure 13 shows the *SIS equipment level* and gives an overview of functional safety related properties and requirements ("data elements") that should be included in the model/shell. In the figure the different properties and requirements have been grouped according to some further defined commonalities (top level). The suggested grouping includes:

A. *Equipment ID & references:* i.e., general attributes, "labels" and references that are required to sufficiently identify the equipment under consideration. This will include information about the associated installation, functional location / tag nr. / NORSOK system number, class, and type of equipment (ref. previous section), equipment boundaries, equipment manufacturer, model specification, and a reference to the SIF(s) that the equipment belongs to. Note that specific equipment location on installation is included as a reliability influencing property but will also normally be included as a part of the equipment ID / functional location.

B. *Reliability Influencing properties (RIPs):* i.e., properties with a potential to impact the reliability performance, here specifically the failure rate, of the equipment under consideration. These RIPs have been further split into:

    i. *Device list of properties (DLOPs),* i.e., aspects used to describe the construction and design of the device, such as dimensions of a valve, measuring/sensing principle of a transmitter, type of internal diagnostics for a gas detector, etc.
    ii. *Operating list of properties (OLOPs),* i.e., aspects related to the operational environment in which the device is used, e.g., type of service/medium, degree of weather exposure, functional application (e.g., ESD and/or PSD) etc.

C. *Safety list of properties (SLOP):* here defined as functional safety and reliability related "*properties characterising the ability of an item to perform a required function under given conditions for a given period of time*"[3]. These properties shall be defined within the context of the functional safety standards to which the SIS equipment shall adhere, i.e., particularly the IEC 61511 (and IEC 61508). This implies that the SLOPs are mainly (but not fully) defined by the SRS (Safety Requirement Specification). Note that whereas the RIPs (DLOPs and OLOPs) mainly affect the *inherent failure rate* of the equipment itself, the SLOPs represent properties with a further impact on the failure on demand probability (PFD). E.g. test interval, diagnostic coverage (DC) and hardware fault tolerance (HFT) are all examples of such properties. The SLOPs can (based on where they mainly apply) be further divided into two classes, design related properties and operation & maintenance related properties[4]:
    i. *Design related,* i.e., safety properties and parameters mainly applicable to the equipment design, e.g. the assumed failure rate from design, fail safe design, stated DC, etc.
    ii. *O&M (Operation and Maintenance) related,* i.e., safety properties and parameters mainly applicable to the operation and maintenance of the equipment, such as test interval, test method and test coverage.

D. *O&M inventory properties:* i.e., properties related to the operational and maintenance history of the equipment under consideration, mostly found in the CMMS (Computerized Maintenance Management System), such as:

    i. *Operational inventory properties,* e.g., service start and end date, time in operation, demand rate, etc.
    ii. *Maintenance inventory properties,* e.g., number and dates of functional tests, test results, findings from other maintenance activities, etc.

E. *Failure history properties:* i.e., properties related to the failure history of the equipment under consideration, such as date of failure, reference to failure notification and/or work order, how the failure was detected, failure mode, etc. Also normally found in the CMMS.

[3] See IEC 61987 CDD: https://cdd.iec.ch/cdd/iec61987/iec61987.nsf/TreeFrameset?OpenFrameSet&ongletactif=1
Also, reference is made to APOS H1 report /14/.

[4] Note that several of the SLOPs defined in design are to be considered as requirements and assumptions for operation, and as such have their equivalent during operation, e.g., assumed test interval from design versus implemented test interval during operation, assumed failure rate from design versus calculated operational failure rate, etc.

The suggested grouping of and associated functional safety properties are illustrated in Figure 13. Note that there may be several approaches for grouping of properties, also depending on use case. Other disciplines will keep other submodels (or other *views* into the information model) as indicated by the grey boxes in Figure 13. As such, other properties, that in some cases can be relevant for functional safety (such as mentioned pressure class for valves) may be available for the user, but from other (discipline) submodels (or by defining alternative *views*). See Chapter 4 for a more detailed discussion.

The groups of functional safety properties may, as indicated in Figure 13, be held as submodels in the SIS equipment admin shell but can alternatively be grouped in one common submodel ("functional safety properties"). This "structuring challenge" is an issue for further consideration and is also briefly discussed in section 3.3.

Some additional comments to Figure 13:

1) A split has been made between (1) data elements where the *list of possible types or values* is general (represented in blue) and applies to all types of SIS equipment, such as failure rate, SIL level, diagnostic coverage, and type of external environment, and (2) data elements where the list of possible types is equipment specific (represented in green), such as sensing principle and failure mode.
2) Other dimensions could also have been highlighted in the figure, e.g., that some data elements by nature are *static*, such as associated equipment class, dimension, and service start date, whereas some data elements are *dynamic* and may change or be revised/updated throughout the equipment life, such as failure rate, time in operation and test frequency.
3) A main document for describing SIF (and SIS component) requirements is the safety requirement specification. The data elements for which information can normally be obtained from the SRS are marked with the symbol * in the figure. Note that in our model (see Figure 12) the SRS is (mainly for illustration) suggested as a separate shell (from where information can be obtained).
4) As discussed earlier, different users may have different views into an AAS. In Figure 13 only functional safety related data elements are shown, but information relevant for other disciplines may be held in the same shell as indicated in the grey branch on the righthand side of the figure and further discussed in chapter 4.
5) As discussed for Figure 12, the admin shells may exist across several lifecycles, but the property values (based on the list of property types as defined in templates and CDDs) will typically be set (for the first time) at different points in time ("instantiation"). E.g., the SIL level may be allocated already during the FEED (Front End Engineering Design) phase, the sensing principle for a specific transmitter may be decided during detail design, whereas the maintenance program with complete overhaul intervals may not be decided until the pre-ops phase. This is indicated by the timeline in the bottom part of the figure (but on a high level and as discussed above many property values may be updated as the project and operation proceeds).
6) It has been questioned whether Figure 13 primarily focuses on field equipment. However, the figure also applies to control logic units, such as logic solvers. This is further described in the APOS H1 report /14/.

Figure 14 shows the content of the information model with properties and requirements held at the SIF level. This includes SIF Id information, properties describing the SIF configuration, requirements from SRS at the SIF level, as well as SIF operational history information.

Some additional comments to Figure 14:

1) Similar as for SIS equipment, a split has been made between (1) data elements where the *list of possible types or values* is general (represented in blue) and applies to all SIFs, such as voting configuration or SIL requirement, and (2) data elements where the list of possible types or values is SIF specific (represented in green), such as SIF description and reference to SIF tags.
2) It is a matter of discussion whether the RBD shall be described and contained in the SIF information model itself or in a separate SIL assessment/verification application. For now, the latter is assumed and included as a reference in Figure 14.
3) A main document for describing SIF (and SIS component) requirements is the safety requirement specification. The data elements for which information can normally be obtained from the SRS are marked with the symbol * in the figure. Note that in our model (see Figure 12) the SRS is described as a separate shell (from where information can be obtained).

For a more detailed discussion of different types of properties and extensive tables of standardised property values, reference is made to the APOS H1 report, /14/.

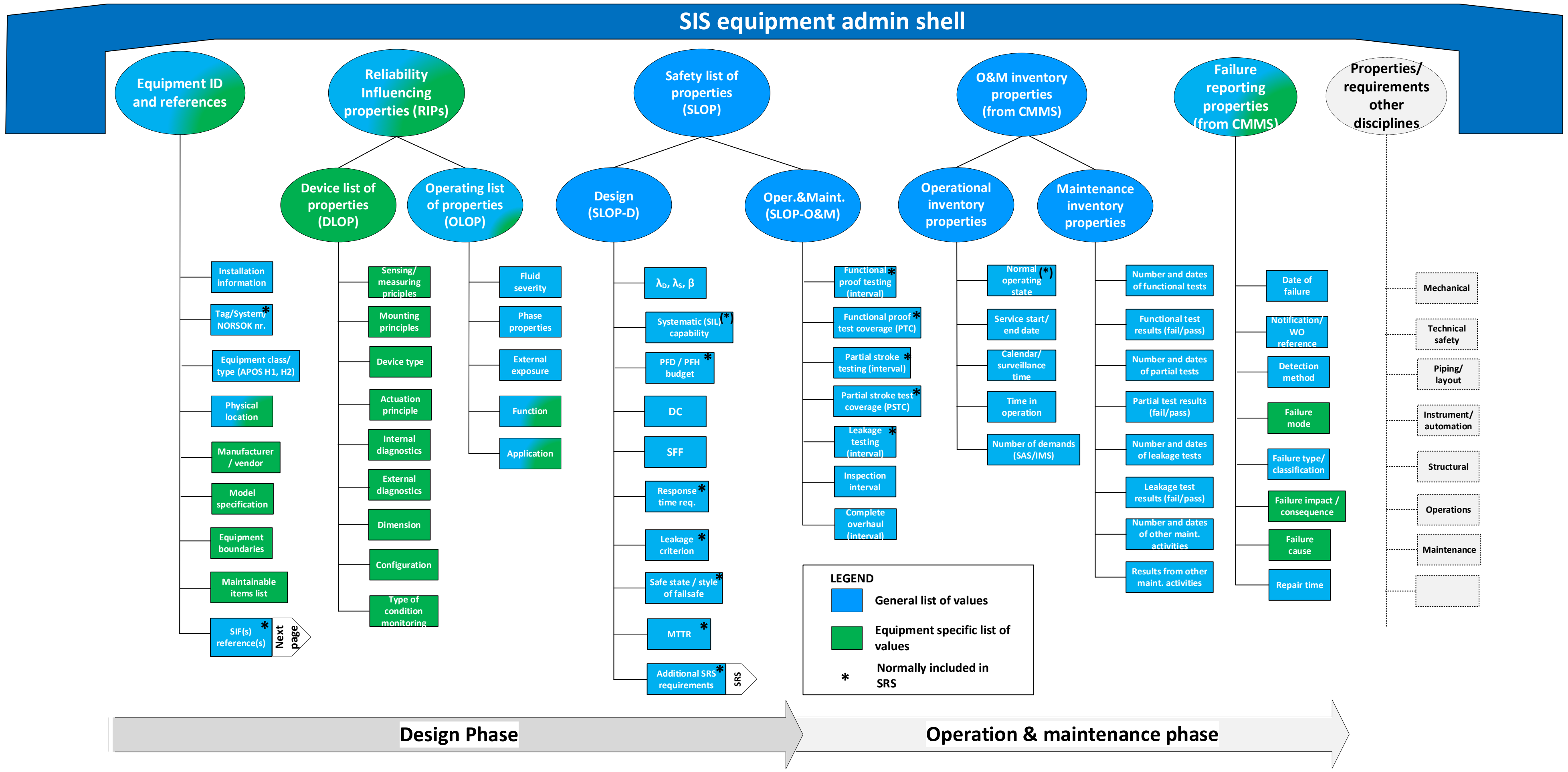

SIS equipment admin shell
Equipment ID and references
Reliability Influencing properties (RIPs)
Safety list of properties (SLOP)
O&M inventory properties (from CMMS)
Failure reporting properties (from CMMS)
Properties/ requirements other disciplines
Device list of properties (DLOP)
Operating list of properties (OLOP)
Design (SLOP-D)
Oper.&Maint. (SLOP-O&M)
Operational inventory properties
Maintenance inventory properties
Installation information
Tag/System/ NORSOK nr. *
Equipment class/ type (APOS H1, H2)
Physical location
Manufacturer / vendor
Model specification
Equipment boundaries
Maintainable items list
SIF(s) reference(s) *
Next page
Sensing/ measuring priciples
Mounting principles
Device type
Actuation principle
Internal diagnostics
External diagnostics
Dimension
Configuration
Type of condition monitoring
Fluid severity
Phase properties
External exposure
Function
Application
λ_D, λ_S, β
Systematic (SIL) capability (*)
PFD / PFH budget *
DC
SFF
Response time req. *
Leakage criterion *
Safe state / style of failsafe *
MTTR *
Additional SRS requirements *
SRS
Functional proof testing (interval) *
Functional proof test coverage (PTC) *
Partial stroke testing (interval) *
Partial stroke test coverage (PSTC) *
Leakage testing (interval) *
Inspection interval
Complete overhaul (interval)
Normal operating state (*)
Service start/ end date
Calendar/ surveillance time
Time in operation
Number of demands (SAS/IMS)
Number and dates of functional tests
Functional test results (fail/pass)
Number and dates of partial tests
Partial test results (fail/pass)
Number and dates of leakage tests
Leakage test results (fail/pass)
Number and dates of other maint. activities
Results from other maint. activities
Date of failure
Notification/ WO reference
Detection method
Failure mode
Failure type/ classification
Failure impact / consequence
Failure cause
Repair time
Mechanical
Technical safety
Piping/ layout
Instrument/ automation
Structural
Operations
Maintenance
LEGEND
General list of values
Equipment specific list of values
* Normally included in SRS
Design Phase
Operation & maintenance phase

Figure 13 Typical content of functional safety information model – SIS equipment level

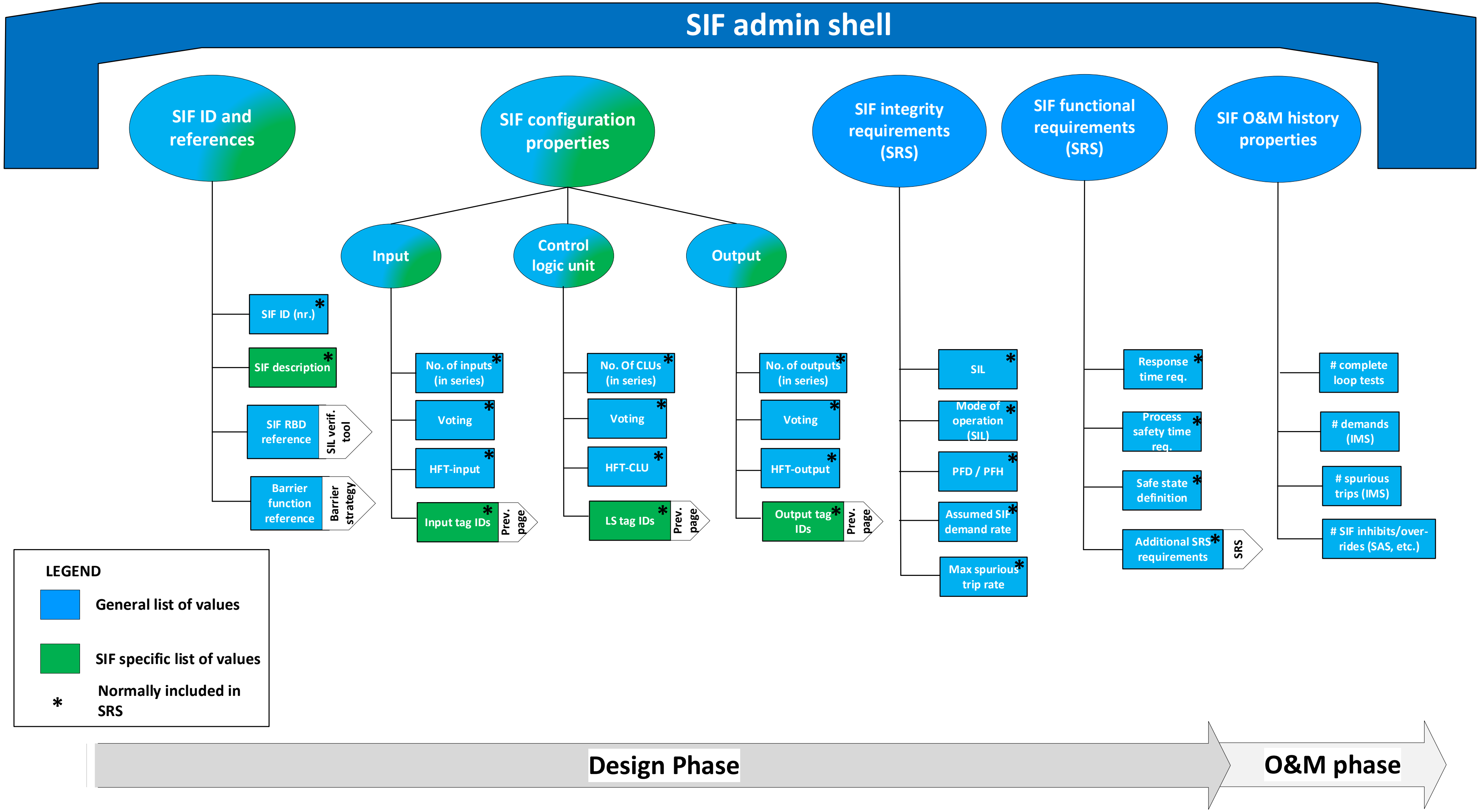


Figure 14 Content of information model - SIF level

3.3 Alignment with asset administration shell (AAS)

Some challenges and benefits of the AAS concept are discussed, and two perspectives may be of particular interest:

1) a lifecycle perspective, i.e., how a functional safety asset like a SIF will exist from specification to engineering throughout operation, and
2) a bottom-up as well as a top-down perspective, i.e., what is the typical content of a SIF asset administration shell (bottom-up), and how does the SIF AAS relate to other assets within functional safety as well as barrier management in general (top-down).

In the following, we will use the term "SIF-AAS" to denote this scope of work.

The international electrotechnical committee (IEC) is in the process of generating AAS standards based on Platform Industry 4.0 specifications through the IEC working group WG 24 (TC 65). As a first step, the standards will be published as technical reports (TRs), which is a faster way to get the documents published. Per August 2022, part 1 of IEC TR 63278 /4/ had the status as Committee draft for voting (CDV) /4/, whereas parts 2 and 3 had status as New Proposals (/6/ and /7/), but the statuses are evolving continuously. Experts involved in the standardization of IEC and within the Industry 4.0 platform are in the meantime sharing ideas, specifications, and implementation examples through various reports, whose status can be found at e.g. the Industry 4.0 platform webpage for Asset Administration Specifications. The most "up to date" status for AAS implementation is therefore found in the more recent AAS reports marked "Industry 4.0 platform". Ongoing discussions within the Industry 4.0 platform reports imply that further work remains to reach a commonly agreed specification that results in a sufficiently contained IEC standard for the modelling and realization of AAS. Also, the two organizations, OPC Foundation and AutomationML, have proposed companion standards that convert AAS information models to OPC UA and Automation ML formats respectively.

The current AAS specifications indicates large flexibility regarding how the digital representations of an asset can be implemented. Some examples are:

- What is to be regarded as an asset is quite flexible with the inclusion of physical, digital, as well as intangible assets. For example, a system integrator or consultancy company may propose that a safety requirement specification (SRS) is a (digital) asset and thereby introduce an SRS AAS shell, while others may prepare an SRS submodel to be integrated into another existing AAS.
- Generally, we observe that the same asset information and asset functions can be realized as self-contained submodels or by their own AAS shells with an underlying submodel structure. For example, one vendor may deliver a set of submodels for a product that can be integrated with an AAS generated by others, while another vendor will deliver a self-contained AAS.
- An asset may be represented by a single AAS shell or as a composite of multiple shells.
- CDD repositories like those based on IEC 61987 /45/, IEC 61360-4 /46/, IEC 62683 /48/, and ISO 15926 / CFIHOS[5] /49/, indicate that different sectors and user groups can suggest their own structure and definitions of properties, even for the same type of components. The intention of having a property or attribute defined one single time in one common repository, using the same naming (e.g. having a unique IRDI (International Registration Data Identifier) code for a push button and its associated properties) is therefore, as of today, not fulfilled. Rather it appears that translation tables or mappings between standards, such as e.g. between IEC 61987 /45/ and ISO 15926 /49/, is currently the most realistic way forward. Within the same submodels or AAS, the

[5] https://www.jip36-cfihos.org/cfihos-standards/

different vendors may therefore use different references for e.g. Safety Integrity Level (SIL) if more than one code or definition is found within the repositories.

This flexibility may imply that different stakeholders will develop different AAS representations that can become challenging to align.

The following topics are therefore suggested for further investigation and discussion among key stakeholders that are to adopt AAS for the process industry:

- Considerations on what should be defined as assets or not
- Considerations of where to use composite AASs
- The balance between AAS and submodels; what should be modelled as separate assets and what should be defined as submodels?
- How to avoid overlap and thus repetition of information between submodels
- Consider whether it is realistic to obtain the goal of having a single reference point and the same naming for identical data elements, or whether mapping tables are required
- How to fully utilize the functionality and possibilities offered by the AAS framework, including aspects such as capabilities, operations, services, resources, relationships, collections, qualifiers, views, etc.
- How the digital infrastructure following AAS implementation will influence the way of working for different stakeholders, and for cooperation between the stakeholders

We have noted that some of the initiatives to define properties and property values for functional safety (e.g. draft amendment to CDD), seem to have chosen a bottom-up approach. Some properties are referring to IEC 61508, while others are based on a sector specific standard. IEC 61508 noticed some years ago that the deviation between generic concepts and sector specific concepts and approaches was increasing. For this reason, the IEC 61508 working group carried out an alignment initiative where the role of IEC 61508 as a horizontal standard was clarified: Unless there were specific rationales within the sector specific standards to deviate from IEC 61508 terms and concepts, the IEC 61508 ones should be used. A similar approach may at some point in time be necessary also for CDD initiatives.

Most of these questions will have to be further explored and clarified in future studies, but some initial thoughts are given below.

Assets, AAS, AAS composites, submodels and ownership

In Industry 4.0, the term asset, being any "object which has a value for an organization" (see Table 1), is of central importance. Thus, assets in Industry 4.0 can take almost any form, for example, be a production system, a product, a software installation, intellectual properties, or even human resources, /3/.

In the context of functional safety, it can be relevant to identify assets within all three categories (see definition in Table 1):

- Physical assets: SIS Equipment and SIFs
- Digital assets: Information collected in reports, databases, and digital drawings
- Intangible assets: Industry 4.0 platform reports suggest software licenses as an example of intangible assets. However, it may be of interest to investigate if this concept is applicable also to

human and organizational resources and possibly specific lifecycle activities such as functional safety management (cf. IEC 61511)

An AAS is a *digital* representation of one or more of these three categories of assets and will, of course, per definition always be digital. At the same time, a physical asset (like a transmitter) can also be complemented by several digital assets related to e.g., documentation (SIL certificates, safety manual, etc.) and software (e.g. license). A relevant question may then arise: Should the three types of assets be represented by *one* single AAS, with the information representing the physical, digital, and intangible assets contained in submodels, or should the three assets have their own AAS in a kind of composite AAS structure? And/or should the AAS to the degree possible provide link to other sources and systems where the digital assets and other relevant information are contained? As for now, it is difficult to point at the best strategy, but this is suggested as a topic to investigate further in future studies.

IEC TR 63278-1 /4/ defines an AAS as a digital representation of an asset and gives examples of where the digital representation consists of a unique admin shell or several admin shells. A motivation for having multiple AASs is that different stakeholders may generate their own AAS for the same asset to cover different services: The manufacturer may provide an AAS as a digital representation of a product delivery, another manufacturer may provide an AAS for a condition-monitoring system that can be added to the product, and an asset owner may introduce an AAS for performance management of the products (and similar product types). As already discussed, some of these examples could also be implemented as submodels.

We suggest exploring pros and cons of structuring related information in separate AASs or in one AAS with different submodels in more detail. Important questions to address in this respect include:

- Who are generating and maintaining the AASs?
- Who are generating and maintaining submodels?
- What is the best balance between the two, considering also optimal structures beyond functional safety?
- What information should be included as part of the AAS (and associated submodels) itself and what information should be represented in terms of links/references to other sources and systems?

Considering the questions raised above, we notice that the German Standardization Roadmap for Industry 4.0 /15/ also identifies the need for establishing interfaces between functional safety and an Industry 4.0 environment like AAS. The roadmap refers to a report based on work in the China-Germany (SINO-German) Standardization Cooperation Commission /16/. Figure 15 is presented in both reports. A novel aspect brought up in /16/ is the need to also consider the safety integrity of communication for data associated with functional safety. The safety integrity is not relevant for all types of data, just those associated with the execution of the safety function. They therefore suggest introducing two types of shells: safety related, and non-safety related. Their approach is a bit unclear, e.g., does it mainly address safety critical communication? Our recommendation is therefore to further explore the actual relevance of this work.

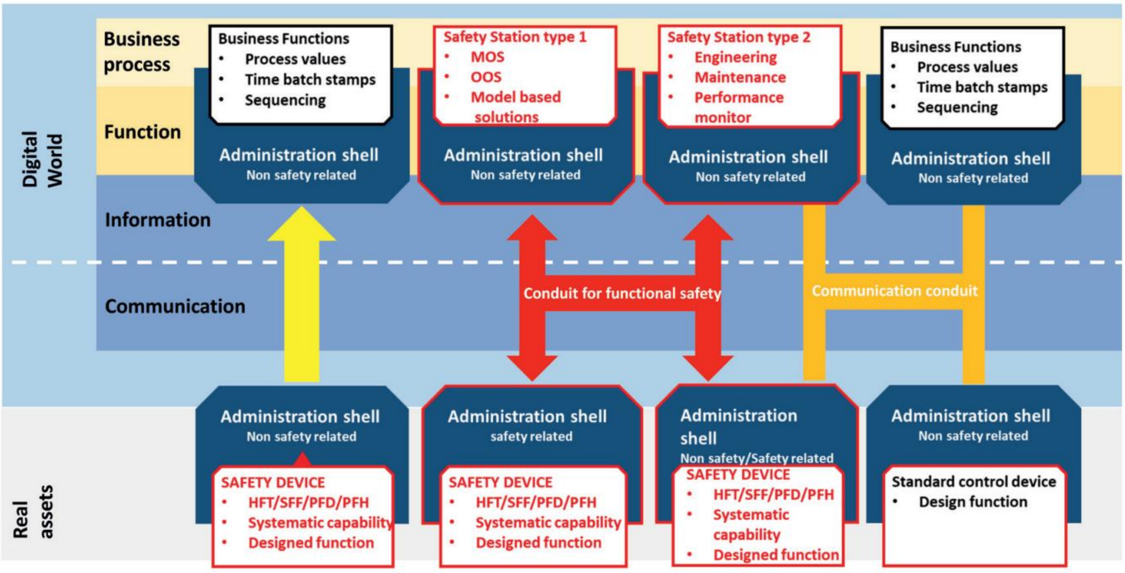


Figure 15 Relating functional safety to the AAS environment (/15/ and /16/)

How to handle updates and changes of specific property values – introducing capabilities and skills

As discussed in relation with Figure 13, some properties have a dynamic nature and may change throughout the lifecycle of an asset. Examples are demand rates, test intervals, failure rates, and even response time requirements and internal leakage criteria that may be changed during operation. As seen from the definitions (Table 1, *note 7* under "property"), a property can have attributes such as code, version, and revision. Hence the AAS has the functionality to manage such changes. It is, however, a question of whether it is more practical to split between design versus operation for some of these dynamic properties. This especially relates to properties (or requirements) where it must be easy to set up compliance queries, e.g.:

- *Design failure rate* versus *operational failure rate*; The design failure rate is an underlying assumption or requirement for SIL compliance and may also change in the design phase as more specific information about the asset arise. The operational failure rate is based on operational failure history. The initial value (at production start-up) is set equal to the final design failure rate and updated (e.g. annually or bi-annually) as operational experience becomes available.
- *Design test interval* versus *operational test interval.* Test interval requirements from design are set in the SRS and shall (at least in theory) correspond to the initial test interval for operation. As operational history is gathered, it may, however, become necessary to change this operational test interval.
- *Assumed demand rate from design* versus *experienced demand rate from operation;* as for test interval, the demand rate is a requirement set in the SRS from the design phase. The demand rate from operation is based on logging/counting of demands from SAS (Safety and Automation System), etc.

The above discussion illustrates the fact that within functional safety it is important to clearly distinguish *requirements* from *achieved performance*. Requirements originate from the design phase, whereas achieved performance shall be monitored during operation.

Requirements and achieved performance may also be regarded as so-called capability statements, either as a specified capability or as an achieved/provided capability. A recent whitepaper by Industry 4.0 on "Describing Capabilities of Industry 4.0 Components" /10/ presents a new extension for the AAS metamodel which satisfies capability-based engineering. The extension introduces new types of submodels for representing capability and skill. Here, the term *capability* is defined as:

> *The implementation-independent description of the function of a resource to achieve a certain effect in the physical or virtual world.*

The submodel for a capability can be defined in two ways: To specify the required capability and the provided capability. In the context of functional safety, we may see the relevance of this opportunity to distinguish performance requirements from actual or estimated performance.

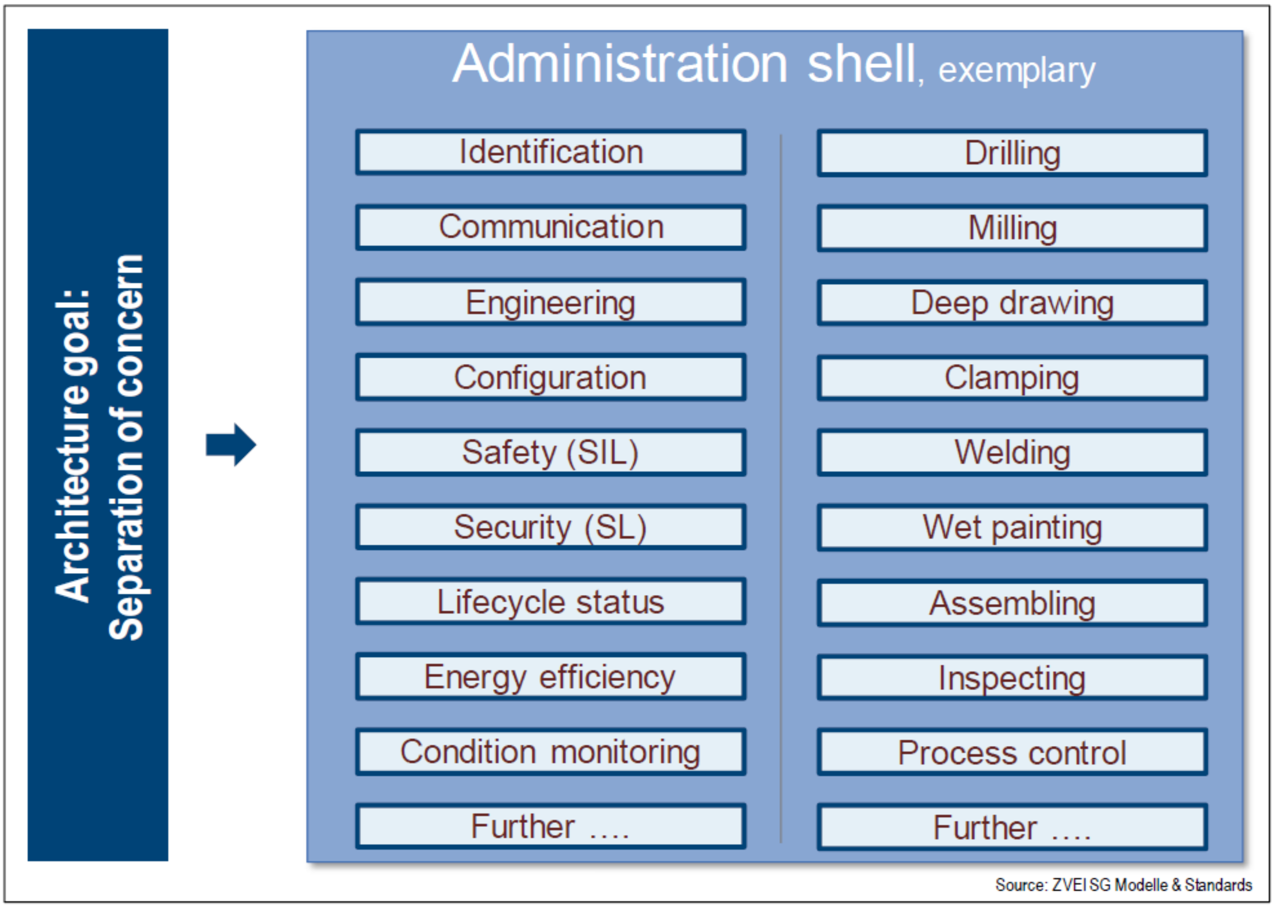
A complementary term to capability introduced in the AAS extension is *skill*, which is defined as:

> *The implementation-dependent description of the function of a resource to achieve a certain effect in the physical or virtual world.*

An important distinction between capability and skill is that a skill identifies the specific resource used to carry out the function. The benefit of the AAS is that properties, data, and content associated with the digital representation is defined as a capability one time and shared and aligned with specific implementation (skills) as needed. Its relevance to functional safety can be exemplified by the distinction between *SIF typicals and unique SIFs*. The new AAS extension suggests how to model the relationship between the two through a relationship element. A last suggestion in the AAS extension, which may be a bit controversial, is to replace the current way of realizing common data directories (CDDs) with more expressive ontologies such as with web ontology language, preferably by the same organizations that publish CDD today. The last point will be pursued in further work.

More on submodels

As discussed in section 3.1, information from different technical domains will be associated with a specific asset and thus, many different properties must be represented in the AAS. To manage all this information, submodels provide a separation of data elements into well-defined domains or subject matters. This is illustrated in Figure 16 (Figure 140 in /3/).



Source: ZVEI SG Modelle & Standards

Figure 16 Examples of domains providing properties for submodels of the Administration Shell, /3/

Here, the Administration Shell is made up of a series of submodels, each representing different aspects of the asset. For our functional safety case, the information specified in Figure 12 - Figure 14 will typically be contained in the "Safety (SIL)" submodel.

Observe that in this example, the submodel containing safety information includes *all* safety information, whereas in Figure 13 we have indicated that the functional safety properties are further divided into more specific submodels. This is a matter of choice, standardisation, and practicalities (including use cases) and needs to be further discussed. We have also identified (at least) two initiatives that need further consideration:

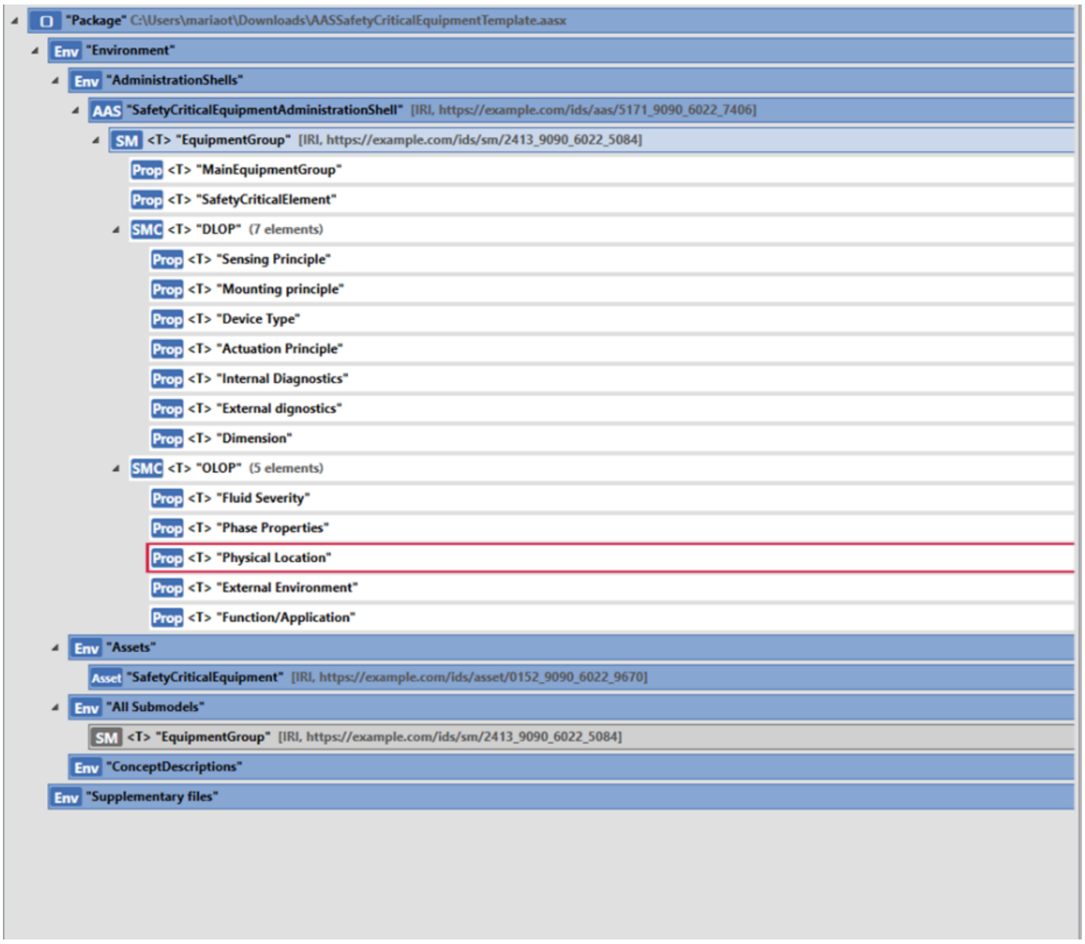
1. The development of submodel templates for specific use cases by the Industrial digital twin association (IDTA) (/17/ and /18/). IDTA (Industrial Digital Twin Association) has started developing submodels for functional safety (and reliability). However, they seem to direct their attention to specific (machinery-related) applications, without addressing the need to build on more generic/horizontal submodels for concepts that can be shared among different sectors.

2. PA-DIM (Process Automation Device Information Model) : We have identified that the FieldComm Group (FCG), OPC Foundation (OPCF) and PROFIBUS/PROFINET International (PI) have taken an initiative referred to as "PA-DIM" to standardize and create information models for process automation devices. Even though this initiative is not addressing functional safety specifically, its relevance is further discussed in chapter 4.

3.4 Preliminary SIS/SIF AAS modelling using AASX Package Explorer

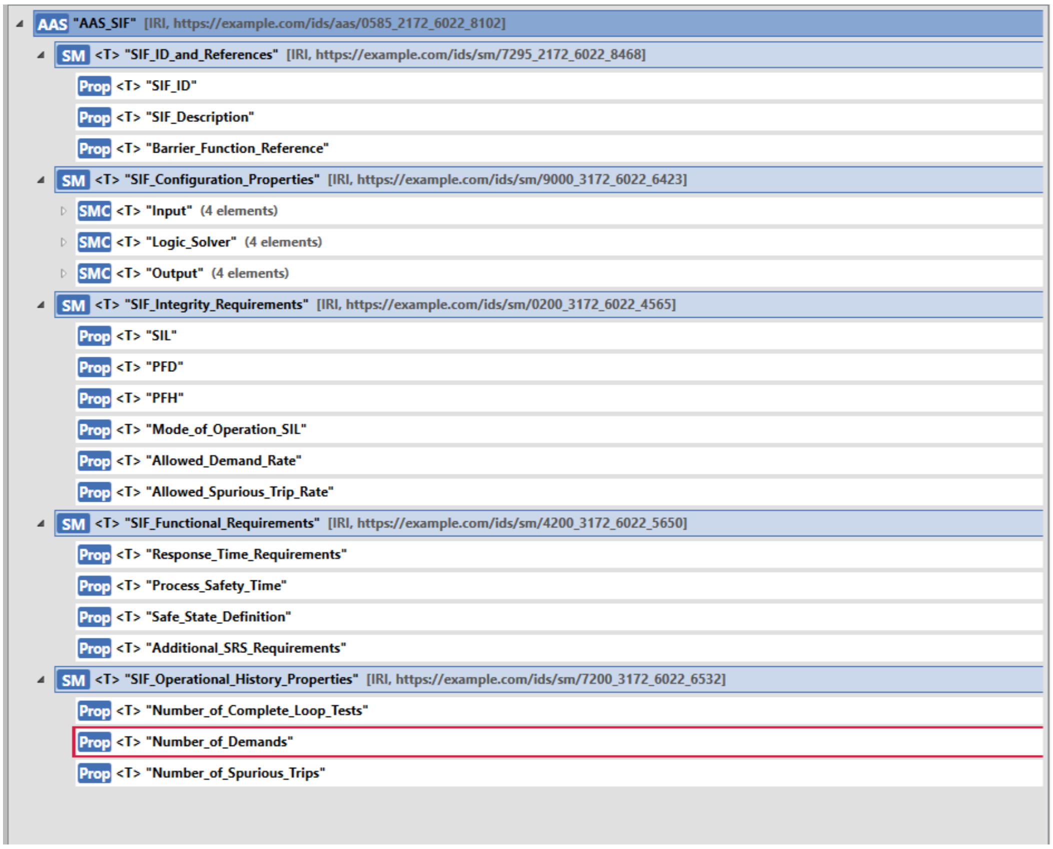
AASX Package Explorer is a C# based open platform tool that helps software developers to work with the Asset Administration Shells.

SINTEF/NTNU have just started looking into this tool, and for the sole purpose of illustration, Figure 17 and Figure 18 show how the equipment and SIF shell and (some of) the properties (see Figure 13 and Figure 14) may be represented in AASX Package Explorer.



```
"Package" C:\Users\mariaot\Downloads\AASSafetyCriticalEquipmentTemplate.aasx
  Env "Environment"
    Env "AdministrationShells"
      AAS "SafetyCriticalEquipmentAdministrationShell" [IRI, https://example.com/ids/aas/5171_9090_6022_7406]
        SM <T> "EquipmentGroup" [IRI, https://example.com/ids/sm/2413_9090_6022_5084]
          Prop <T> "MainEquipmentGroup"
          Prop <T> "SafetyCriticalElement"
          SMC <T> "DLOP" (7 elements)
            Prop <T> "Sensing Principle"
            Prop <T> "Mounting principle"
            Prop <T> "Device Type"
            Prop <T> "Actuation Principle"
            Prop <T> "Internal Diagnostics"
            Prop <T> "External dignostics"
            Prop <T> "Dimension"
          SMC <T> "OLOP" (5 elements)
            Prop <T> "Fluid Severity"
            Prop <T> "Phase Properties"
            Prop <T> "Physical Location"
            Prop <T> "External Environment"
            Prop <T> "Function/Application"
    Env "Assets"
      Asset "SafetyCriticalEquipment" [IRI, https://example.com/ids/asset/0152_9090_6022_9670]
    Env "All Submodels"
      SM <T> "EquipmentGroup" [IRI, https://example.com/ids/sm/2413_9090_6022_5084]
    Env "ConceptDescriptions"
  Env "Supplementary files"
```

Figure 17 AAS package Explorer - Equipment properties



```
AAS "AAS_SIF" [IRI, https://example.com/ids/aas/0585_2172_6022_8102]
  SM <T> "SIF_ID_and_References" [IRI, https://example.com/ids/sm/7295_2172_6022_8468]
    Prop <T> "SIF_ID"
    Prop <T> "SIF_Description"
    Prop <T> "Barrier_Function_Reference"
  SM <T> "SIF_Configuration_Properties" [IRI, https://example.com/ids/sm/9000_3172_6022_6423]
    SMC <T> "Input" (4 elements)
    SMC <T> "Logic_Solver" (4 elements)
    SMC <T> "Output" (4 elements)
  SM <T> "SIF_Integrity_Requirements" [IRI, https://example.com/ids/sm/0200_3172_6022_4565]
    Prop <T> "SIL"
    Prop <T> "PFD"
    Prop <T> "PFH"
    Prop <T> "Mode_of_Operation_SIL"
    Prop <T> "Allowed_Demand_Rate"
    Prop <T> "Allowed_Spurious_Trip_Rate"
  SM <T> "SIF_Functional_Requirements" [IRI, https://example.com/ids/sm/4200_3172_6022_5650]
    Prop <T> "Response_Time_Requirements"
    Prop <T> "Process_Safety_Time"
    Prop <T> "Safe_State_Definition"
    Prop <T> "Additional_SRS_Requirements"
  SM <T> "SIF_Operational_History_Properties" [IRI, https://example.com/ids/sm/7200_3172_6022_6532]
    Prop <T> "Number_of_Complete_Loop_Tests"
    Prop <T> "Number_of_Demands"
    Prop <T> "Number_of_Spurious_Trips"
```

Figure 18 AAS Package Explorer - SIF Properties

4 Interfaces between functional safety and other disciplines

Figure 13 ("Content of information model – SIS equipment level") indicates that as part of an asset administration shell for a SIF component, e.g., a transmitter, there will, in addition to properties and requirements relevant for functional safety, be several non-safety relevant properties, that can be contained within the same shell, but possibly other submodels. Examples of such "other discipline information" can be weight, colour/paint, material of body, mounting details, cable specifications, power supply details, dimensions, removal clearance, etc. As such, functional safety represents a use case that requires a set of specific functional safety related submodels (or possibly one single submodel).

In this chapter the interfaces and integration between information models for functional safety assets and relevant information (models) from other disciplines (process, mechanical, electrical, etc.) are further discussed. This is particularly related to how information structuring may be managed, considering standards for device information models adopted by Industry 4.0. In this context, we will start by introducing field device integration (FDI) and its extension PA-DIM.

4.1 Field device integration (FDI)

Field device integration (FDI) is a technology developed as a joint industry effort with the aim to ease field devices integration and data exchange in a network. The FDI technology was published as a series of IEC 62769 in 2015 and most of the parts have been updated in 2020/21. FDI originally adopted EDDL[6] (Electronic Device Description Language) for describing the behaviour of the field devices (described in IEC 61804) and Field device tool[7] technologies (described in IEC 62453) for realising servers and clients, including information models and interfaces. In the more recent updates, OPC UA is introduced as a platform to realize all these functionalities.

Of most interest for AAS is the FDI information model for devices. One approach to explain the FDI information model is to distinguish between “how to do” and “what to include”. The “how to do” explains how FDI can implement information models and interfaces with devices applying commonly used protocols. Examples include:

- IEC 62769 has dedicated parts with specifications for how commonly used field buses and ethernet-based field communication can interface FDI. The use of OPC UA with FDI is described in IEC 62541-7.
- The OPC UA building blocks of the information models are described in IEC 62769-5. There is also another standard, IEC 62541-100, that provides a more general information model approach for devices. All OPC UA companion standards developed by the OPC foundation relating to device integration and field device integration are listed here: https://opcfoundation.org/developer-tools/documents/ (search “device integration”[8]).

The ”what to include” is perhaps best explained by referring to the field device information model **PA-DIM**. PA-DIM is an information model for field devices that combines the OPC UA realization of FDI technologies *and a* set of NAMUR (German: Interessengemeinschaft Automatisierungstechnik der Prozessindustrie e.V., Eng: international user association of automation technology and digitalization process industries) standards that cover what type of data to share.

[6] EDDL: Electronic device description language. Specified in IEC 61804

[7] Field device tool: Field device tool comprising FDI server, client and information model specified in IEC 62453

[8] The search identifies the following companion standards (by number): 10000-100, 30080-5, 30080-7. 30080 (all parts), 30090.

- NE (NAMUR Recommendation) 131 on field devices proposes which parameters are needed to share the most important functionalities. NE 131 is a bit difficult to read as it mixes German and English language in tables where parameters are specified.
- NE 107 on self-monitoring and diagnostics provides a list of status signals and malfunctions to include for various types of field devices

PA-DIM also builds on the NAMUR Recommendation NE 175 for secure sharing of data from field to cloud.

4.2 Process automation device information model (PA-DIM)

The OPC UA companion standard (30081) contains details about the PA-DIM information model for field devices. Unfortunately, membership in the OPC foundation is required to access the standard, and currently there is no IEC (or ISO) standard available that explains the content in more detail. However, the whitepaper by FieldComm Group /33/ gives an overview of the suggested grouping of data and information for field device data according to the OPC UA companion standard for the PA-DIM information model:

- Device Identification (ID/nameplate: Static data that identifies the field device)
- Device health
- Process values
- Device core parameters (per device profile)

The same document /33/ discusses briefly how PA-DIM can be aligned with the AAS framework. In short, it is proposed that each of the four groups are realized as submodels, as shown in Figure 19.

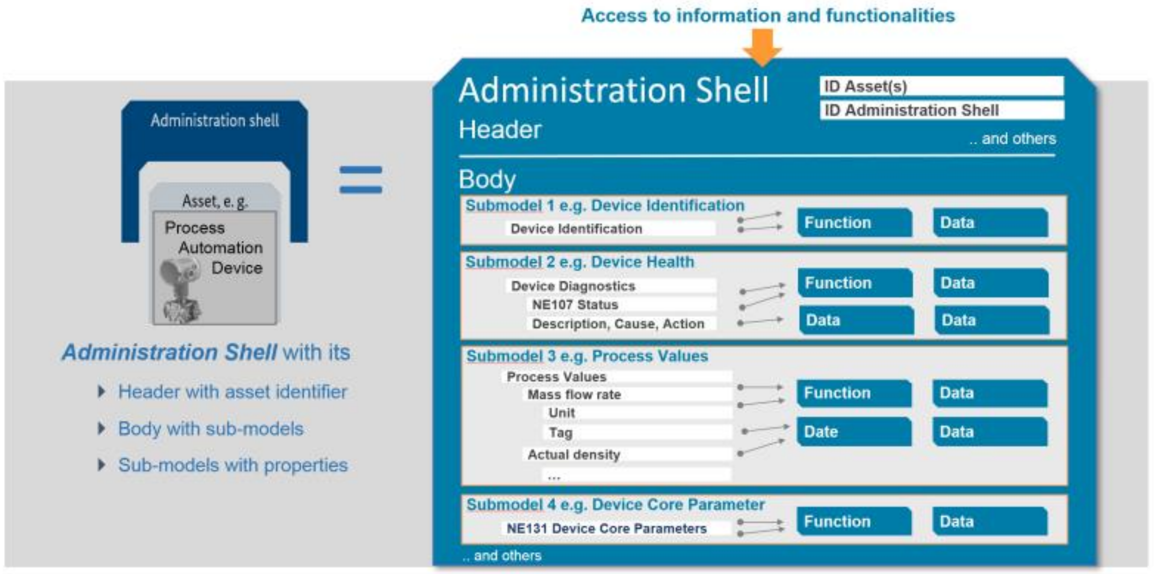


Figure 19 Asset administration shell for process automation devices [/33/]

4.3 Complementing process automation device information model (PA-DIM) for functional safety

At the device level, it seems reasonable to add functional safety-related information models to the PA-DIM information models in a common asset administration shell for the device. However, it needs to be further explored if *all* functional safety data elements, i.e. properties, parameters, parameter values, etc, should be organized in functional safety submodels or if a mapping is needed to identify any overlaps between data in the four PA-DIM (to be) submodels. One can foresee that only data elements relevant to functional safety, but not covered by the four PA-DIM models, are included in the functional safety submodels. The opportunity to access all functional safety related data elements for a specific device may be achieved by using the AAS metamodel for views. According to the Platform Industry 4.0 report on details of the administration shell (part 1) /3/ (p.80-81), the view attribute is added for the following purpose:

> *The large number of submodel elements within a submodel can be filtered by views, so that different user groups can only see relevant elements. A view is a collection of referable elements with respect to a specific viewpoint of one or more stakeholders*

One way to visualize the application of the view feature in AAS with device submodels is illustrated in Figure 20. Here, it is suggested that the general PA-DIM submodels for PA-DIM are complemented by (so far unnamed) functional safety submodels. To avoid replicating data elements in submodels, it is suggested to implement a view functionality in the information model, which can be presented in an API (Application Programming Interface) for functional safety (named “Functional safety view” in the figure).

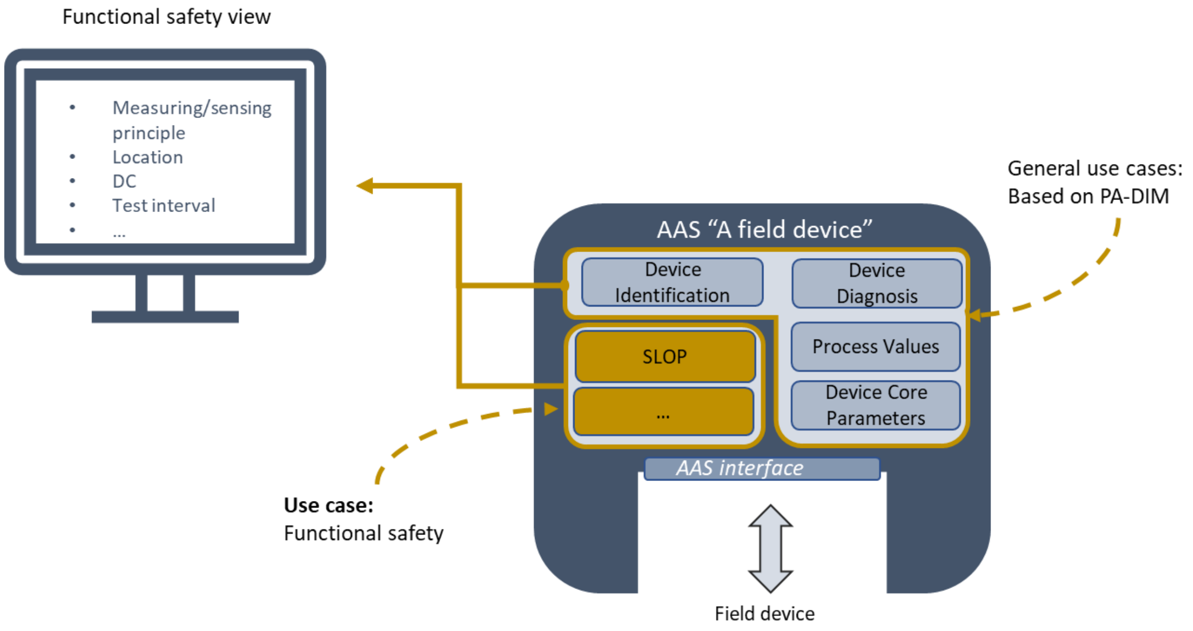


Figure 20 Combining submodels to generate functional safety view

5 Conclusions and further work

A main objective of the work in this report has been to contribute toward further digitalisation of the petroleum industry, by exploring what to include in an information model for functional safety and how such a model can be structured and maintained throughout its lifecycle. In addition, the report attempts to clarify some terms and expressions commonly used within the digitalisation and information modelling domains that are not necessarily well-known for people working with automation and functional safety. Furthermore, the report also discusses how a functional safety information model, e.g., for a typical safety instrumented function (SIF) fits into a bigger picture where interaction and integration with other disciplines' information models are required.

While working with this report, some high-level questions that need further consideration have emerged:

(1) how lifecycle management of functional safety most effectively can utilize the ideas of a digital twin according to the AAS ecosystem.

(2) how to obtain interoperability in the value chain including engineering, operation, reliability performance, and governance reporting by applying the ideas of Industry 4.0 / AAS.

(3) how AAS and its submodels shall be mapped to other industry-relevant standards and technologies, in particular AML and OPC UA.

(4) how could lifecycle management of functional safety within the AAS framework most effectively be integrated with:

   a. the more overarching barrier management activities (including non-instrumented barriers, performance standards, etc.),
   b. Other discipline activities and information needs (instrument, automation, mechanical, structural, etc.).

Some more specific questions and topics to further address include:

- What are to be regarded as assets and when are assets to be represented by single AASs versus composite AASs?
- What is the best way of structuring the AAS and the submodels and what is the best balance between the two, also taking into consideration optimal structures beyond functional safety, multiple user information access, the lifecycle perspective, etc.?
- Who is generating and maintaining the AASs and the submodels in a lifecycle perspective?
- How to fully utilize the functionality and possibilities offered by the AAS framework, including aspects such as capabilities, skills, operations, services, resources, relationships, collections, qualifiers, views, links/references, etc.?
- How to limit repetition of information across several AASs and submodels (single source of information truth)?
- How the digital infrastructure following AAS implementation will influence the way of working for different stakeholders, and for cooperation between the stakeholders?

In addition, there are several topics related to a more detailed SIF use case that needs further investigation, for instance:

1. how to structure the composite AAS(s) for one or more typical SIFs (including submodels)?

2. the information/data content for each AAS and associated submodels, and how (and from who and when) this information becomes available during evolving lifecycles phases, thereby highlighting different AAS content for different lifecycle phases.

3. how the different actors (manufacturers, suppliers, integrators, and operators) contribute with different AAS information throughout different lifecycle phases, including a discussion of the source systems/documents from where this information originates, thereby highlighting possibilities and challenges related to interoperability in the value chain.

4. the difference between and handling of type assets versus instance assets as well as submodel types and submodel templates and related roles (e.g., AAS responsible).

Although further work on these topics will take a *functional safety perspective*, information needs from other disciplines must also be considered, since admin shells and/or submodels will embrace several disciplines (see e.g. Figure 16 and Figure 20).

6 References

/1/ IEC 61511 (2016). Functional safety - safety instrumented systems for the process industry sector. Part 1 – 3, Edition 2

/2/ IEC 61508 (2010). Functional safety of electrical/electronic/ programmable electronic (E/E/PE) safety related systems. Part 1-7, Ed. 2.0.

/3/ Details of the Asset Administration Shell, Part 1 - The exchange of information between partners in the value chain of Industry 4.0 (Version 3.0RC01), November 2020

/4/ IEC TR 63278-1. Asset Administration Shell for industrial applications – Part 1: Asset Administration Shell structure, Committee draft for voting (CDV), circulated May 2022

/5/ IEC TR 63278-2. Asset Administration Shell for industrial applications – Part 2: Information Meta Model, New work item proposal (NP), Circulated Jan 2022

/6/ IEC TR 63278-3. Asset Administration Shell for industrial applications – Part 3: Security Provisions for Asset Administration Shells, New work item proposal (NP), Circulated Jan 2022

/7/ Usage View of Asset Administration Shell. Technical report by Platform Industrie 4.0 (https://www.researchgate.net/publication/331959412_Usage_View_of_Asset_Administration_Shell)

/8/ Industrie 4.0 Plug-and-Produce for Adaptable Factories. Examples of Use Case Definition, Models, and Implementation. Working paper from Platform Industrie 4.0. https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2017/Juni/Industrie_4.0_Plug_and_produce/Industrie-4.0-_Plug-and-Produce-zvei.pdf

/9/ Wallner, R. and Lundteigen, M.A. *Approaches to utilize Digital Twins in Safety Demonstration and Verification of Automated and Autonomously Controlled Systems*. Paper to be presented at ESREL 2022, Dublin, 28-1.9.2022.

/10/ Describing Capabilities of Industrie 4.0 Components. White paper from Platform Industrie 4.0. Nov 2020. Available from; https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Capabilities_Industrie40_Components.html

/11/ IEC 62541-100 OPC Unified Architecture – Part 100: Device Interface (2015)

/12/ NAMUR NE 107 Self-Monitoring and Diagnosis of Field Devices (2017)

/13/ NAMUR NE 131 NAMUR standard device – Field devices for standard application (2017)

/14/ Hauge, S., Håbrekke, S., et al (2023), Guidelines for standardised classification and failure reporting for safety equipment in the petroleum industry (APOS H1), SINTEF Report no. 2023:00108

/15/ German Standardizaton Roadmap Industrie 4.0 (version 4, March 2020). Access from here: https://www.din.de/resource/blob/65354/1bed7e8d800cd4712d7d1786584a7a3a/roadmap-i4-0-e-data.pdf

/16/ Sino-German White Paper on Functional Safety for Industrie 4.0 and Intelligent Manufacturing (July 2020). Accessed from here: https://www.bmwk.de/Redaktion/DE/Publikationen/Industrie/industrie-4-0-sino-german-white-paper-on-functional-safety-for-industry-4-0-and-intelligent-manufacturing.pdf?__blob=publicationFile&amp;v=16

/17/ Use Cases – the Digital Twin in Practice. Access from here: https://industrialdigitaltwin.org/en/use-cases by Industrial Digital Twin Association (IDTA). Accessed 24.08.2022.

/18/ AAS Submodel Templates. Access from here: https://industrialdigitaltwin.org/en/content-hub/submodels by Industrial Digital Twin Association (IDTA). Accessed 24.08.2022.

/19/ Grieves, M. (2015, 03). Digital twin: Manufacturing excellence through virtual factory replication. Whitepaper, 1–7.

/20/ Kritzinger, W., M. Karner, G. Traar, J. Henjes, and W. Sihn (2018). Digital twin in manufacturing: A categorical literature review and classification. IFACPapersOnLine 51(11), 1016–1022. 16th IFAC Symposium on Information Control Problems in Manufacturing, INCOM 2018

/21/ Bratbak, E., Asset Administration Shell for Life Cycle Management of Safety systems, Master's thesis in Cybernetics and Robotics, March 2022. Accessed from here: https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2997101/no.ntnu%3Ainspera%3A91918311%3A45135875.pdf?sequence=1

/22/ IEC 62541-5 OPC Unified Architecture – Part 1: Overview and concepts (2020)

/23/ IEC 62541-5 OPC Unified Architecture – Part 5: Information Model (2020)

/24/ OPC 30270 OPC UA for Asset Administration Shell (AAS) (2021)

/25/ IEC 62714-1 Engineering data exchange format for use in industrial automation systems engineering - Automation Markup Language - Part 1: Architecture and general requirements (2018)

/26/ IEC 62424 Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools (2016)

/27/ ISO 17506 Industrial automation systems and integration — COLLADATM digital asset schema specification for 3D visualization of industrial data (2022)

/28/ AutomationML White Paper, Part 1 – Architecture and General Requirements version 2.1 (2018)

/29/ AutomationML Application Recommendation: AAS Representation version 1.0 (2019)

/30/ Schönwälder, J., Information Models, Data Models, and YANG, IETF 86, Orlando, 2013-03-14

/31/ ISO 14224 (2016), Petroleum, petrochemical and natural gas industries — Collection and exchange of reliability and maintenance data for equipment.

/32/ Common Data Dictionary for process industry (based on IEC 61987). Accessed from here: https://cdd.iec.ch/cdd/iec61987/iec61987.nsf/TreeFrameset?OpenFrameSet&ongletactif=1

/33/ FieldComm Group. Process Automation Device Information model. 2019. Access from here : https://www.fieldcommgroup.org/sites/default/files/imce_files/technology/documents/PA%20DIM%20white%20paper%201.0.pdf

/34/ RAMI4.0 – a reference framework for digitalisation, https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.pdf?__blob=publicationFile&v=3

/35/ https://www.researchgate.net/publication/334429449_Specification_Demonstrator_I40-Language_v30

/36/ IDTA 02014-1-0 Functional safety for safety-relevant devices. 2022. Accessed from here: https://github.com/admin-shell-io/submodel-templates/blob/main/published/Functional%20Safety/1/0/IDTA%2002014-1-0_Submodel_FunctionalSafety.pdf

/37/ ECLASS - Global reference data standard for the classification and unambiguous description of products and services, Accessed from here: https://eclass.eu/en/eclass-standard.

/38/ Wright, L., Davidson, S.; (2022). How to tell the difference between a model and a digital twin, Advanced Modeling and Simulation in Engineering Sciences 7(1).

/39/ AutomationML. Accessed from here: https://www.automationml.org/

/40/ JSON. Accessed from here: https://www.json.org/json-en.html

/41/ Eclipse BaSyx. Accessed from here: https://www.eclipse.org/basyx/

/42/ AASx Package Explorer, Accessed from here https://github.com/admin-shell-io/aasx-package-explorer

/43/ Hansen, G.K, Onshus, O., Jaatun, M.G., Myklebust, T., Ottermo, M., Lundteigen, M.A. (2021), Principles of digitalisation and IT-OT integration. Accessed from here: https://www.ptil.no/globalassets/fagstoff/prosjektrapporter/ikt-sikkerhet/sintef---report---principles-of-digitalisation-and-it-ot-integration.pdf

/44/ IEC 62714, Engineering data exchange format for use in industrial automation systems engineering - Automation Markup Language

/45/ IEC 61987-1 (2006), Industrial-process measurement and control - Data structures and elements in process equipment catalogues

/46/ IEC 61360, Standard data element types with associated classification scheme.

/47/ ISO 13584-42 (2010), Parts library

/48/ IEC 62683-1 (2017), Low-voltage switchgear and control gear - Product data and properties for information exchange - Part 1: Catalogue data

/49/ ISO 15926, Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities

/50/ MQTT. Accessed from here: https://mqtt.org/

/51/ Håbrekke, S., Hauge, S. and M. A. Lundteigen (2023), Guideline for follow-up of Safety Instrumented Systems (SIS) in the operating phase, Ed. 3 (APOS H3), SINTEF Report no. 2023:00107

/52/ Lee, S., M. V. Ottermo, S. Hauge, S. Håbrekke, M. A. Lundteigen (2023), Potential for automated follow-up of safety equipment (APOS H2), SINTEF Report no. 2023:00110

/53/ Ottermo, M.V., Hauge S. and Håbrekke S., (2021). Reliability Data for Safety Equipment, PDS Data Handbook. SINTEF Report 2021:00370, ISBN 978-82-14-06468-1.

/54/ Hauge, S., Kvam, E. (Safetec) and APOS H4 working group, Specification for standardised electronic SRS (APOS H4 project memo), March 2023.

SINTEF

Technology for a better society

www.sintef.no