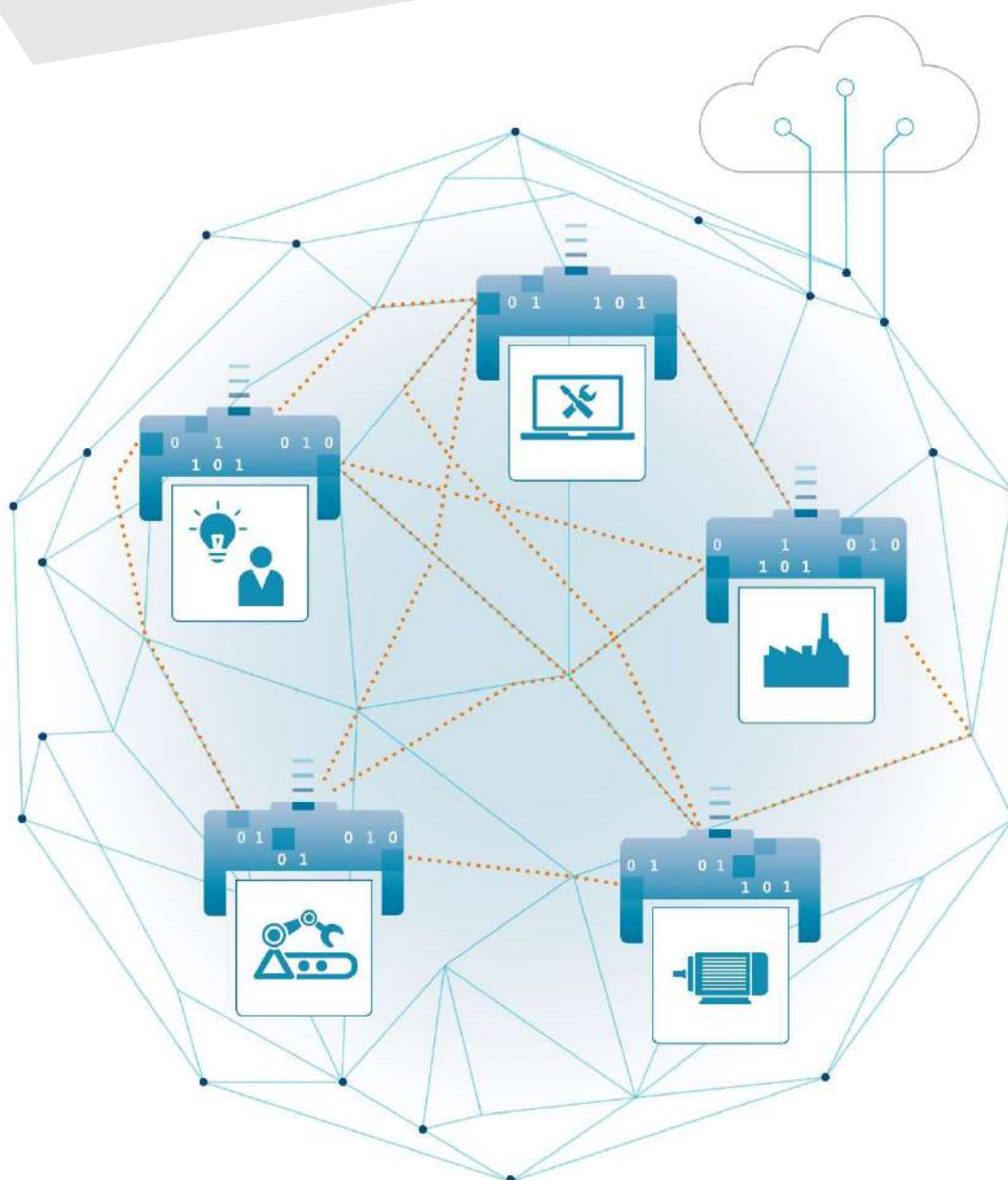


ESPECIFICACIÓN

Los detalles de la Concha de  
Administración de Activos



Parte 1 - El intercambio de información entre los  
socios en la cadena de valor de Industrie 4.0 ( Versión  
1.0)

# Imprimir

## Editor

Ministerio Federal de Economía y Energía (BMWi)  
Relaciones Públicas 10119 Berlín [www.bmwi.de](http://www.bmwi.de)

## Texto y edición

Plattform Industrie 4.0 Bertolt  
Brecht-Platz 3 10117 Berlín

## Diseño y producción

La secretaría Plattform 4.0 Industrie, Berlín

## Estado

de noviembre de 2018

## Ilustraciones

Plattform Industrie 4.0; Anna Salari, diseñado por Freepik (Título)

## Contenido

|       |   |       |
|-------|---|-------|
| 1     | Preámbulo .....   | 10    |
| 1.1   | notas editoriales .....   | 11    |
| 1.2   | Alcance de este documento .....   | 11    |
| 1.3   | Estructura del documento .....  | 11    |
| 1.4   | Principios de la obra .....   | 12    |
| 1.5   | Términos y Definiciones .....   | 12    |
| 1.6   | Abreviaturas .....  | 17    |
| 2     | Conceptos básicos y la imagen que lleva .....                                       | 18    |
| 2.1   | Conceptos básicos .....   | 19    |
| 2.2   | Que lleva la imagen .....   | 19    |
| 3     | El metamodelo de la Concha de Administración .....                                  | 22 .. |
| 3.1   | Introducción .....  | 23    |
| 3.2   | Tipos e instancias .....  | 23    |
| 3.2.1 | Ciclo vital .....   | 23    |
| 3.2.2 | Ejemplo .....   | 24    |
| 3.2.3 | Metamodelo de los tipos e instancias de Shell Administración de Activos .....       | 25    |
| 3.3   | Identificación de entidades .....   | 26    |
| 3.3.1 | Visión general .....  | 26    |
| 3.3.2 | Lo Identificadores existir? .....   | 26    |
| 3.3.3 | Identificadores para los activos y las cáscaras de la Administración .....          | 27    |
| 3.3.4 | ¿Qué Identificadores de usar para los cuales las entidades .....                    | 28    |
| 3.3.5 | ¿Cómo se crean nuevos identificadores? .....  | 29    |
| 3.3.6 | Las mejores prácticas para la creación de identificadores URI .....                 | 30    |
| 3.3.7 | Creación de una instancia submodelo basado en un tipo de sub-modelo existente ..... | 31    |
| 3.3.8 | Se pueden formar nuevos submodelos o de propiedad? .....                            | 31    |
| 3.3.9 | El uso de ID corta para las entidades identificables .....                          | 32    |
| 3.4   | Descripción general del metamodelo de la Concha de Administración .....             | 35    |
| 3.5   | Metamodelo detalles Especificación: Los designadores .....                          | 36    |
| 3.5.1 | Introducción .....  | 36    |
| 3.5.2 | atributos comunes .....   | 36    |
| 3.5.3 | Shell de administración de activos Atributos .....                                  | 44    |
| 3.5.4 | Atributos de activos .....  | 45    |
| 3.5.5 | Submodelo y Element submodelo Atributos .....                                       | 46    |
| 3.5.6 | Resumen de los tipos de elemento submodelo .....                                    | 50    |
| 3.5.7 | Elemento de datos de atributos .....  | 50    |

## 4 | PREÁMBULO

|        |   |       |
|--------|---|-------|
| 3.5.8  | Colección de elementos de datos de atributos .....                                    | 52    |
| 3.5.9  | Relación Atributos .....  | 53    |
| 3.5.10 | Operación Atributos .....   | 53    |
| 3.5.11 | Ver atribuye .....  | 54    |
| 3.5.12 | Concepto Diccionario Atributos .....  | 55    |
| 3.5.13 | Hacer referencia en los depósitos de la Administración de Activos .....               | 56    |
| 3.5.14 | Tipos .....   | 57    |
| 3.5.15 | Plantillas, herencia, Calificadores y Categorías .....                                | 59    |
| 3.6    | plantillas predefinidas de especificación de datos .....                              | 60    |
| 3.6.1  | Concepto de plantillas de especificación de datos .....                               | 60    |
| 3.6.2  | Plantillas predefinidas para Descripciones de las propiedades .....                   | 60    |
| 4      | Asignaciones a los formatos de datos para compartir información I4.0 compatible ..... | 62    |
| 4.1    | ..... general .....   | 63    |
| 4.2    | Reglas generales .....  | 63    |
| 4.3    | ejemplo Unified .....   | 64    |
| 4.4    | XML .....   | 66    |
| 4.4.1  | ..... general .....   | 66    |
| 4.4.2  | Introducción .....  | 66    |
| 4.4.3  | Reglas .....  | 66    |
| 4.4.4  | Ejemplo de estructuras de nivel superior .....  | 67    |
| 4.4.5  | XSD grupos de modelos .....   | 68    |
| 4.4.6  | Teclas y Referencias .....  | 69    |
| 4.4.7  | Administración de activos de Shell Mapeo .....  | 69    |
| 4.4.8  | ConceptDescriptions y EmbeddedDataSpecifications Mapeo .....                          | 70    |
| 4.5    | JSON .....  | 72    |
| 4.5.1  | ..... general .....   | 72    |
| 4.5.2  | Reglas .....  | 72    |
| 4.5.3  | Ejemplo de estructuras de nivel superior .....  | 72    |
| 4.5.4  | Ejemplos de referencias a identificables .....  | 73    |
| 4.5.5  | Ejemplos para las descripciones .....   | 74    |
| 4.5.6  | Ejemplos para ReferenceElement .....  | 74    |
| 4.5.7  | Ejemplos para GlobalReference .....   | 76 .. |
| 4.5.8  | Ejemplo para un tipo = "tipo" Referencia .....  | 77    |
| 4.6    | RDF .....   | 78    |
| 4.7    | OPC UA .....  | 78    |
| 4.8    | AutomationML .....  | 78    |
| 5      | Atributo de acceso basado en roles y la base .....                                    | 80    |
| 5.1    | Permisos de Acceso para pasar .....   | 81    |

|          |   |            |
|----------|---|------------|
| 5.1.1    | Acceso efectivo basado en las reglas de permiso de acceso .....   | 81         |
| 5.2      | Filtrado de la Información en la exportación y la importación .....                                     | 82         |
| 5.3      | Shell Resumen de la administración de activos para la Seguridad Metamodel .....                         | 83         |
| 5.4      | <b>Metamodelo detalles Especificación: Los designadores .....</b>                                       | <b>86</b>  |
| 5.4.1    | Introducción .....  | 86         |
| 5.4.2    | ..... común .....   | 86         |
| 5.4.3    | Atributos de seguridad .....  | 87         |
| 5.4.4    | Acceso Política de Control de Punto Atributos .....   | 87         |
| 5.4.5    | Control de Acceso Local Atributos .....   | 89         |
| 5.4.6    | Los atributos de la regla de permisos de acceso .....   | 91         |
| 6        | <b>Paquete Formato de archivo para el Shell de administración de activos (AASX) .....</b>               | <b>94</b>  |
| 6.1      | ..... general .....   | 95         |
| 6.2      | La selección del formato de referencia para el formato de paquete Shell Administración de Activos ..... | 95         |
| 6.3      | Conceptos básicos de las Convenciones de empaquetado abierto .....                                      | 96         |
| 6.4      | Convenciones para el formato de archivo del paquete de Shell Administración de Activos (AASX) .....     | 96         |
| 6.5      | modelo lógico .....   | 97         |
| 6.6      | Modelo físico .....   | 98         |
| 6.7      | Firmas digitales .....  | 101        |
| 6.8      | Cifrado .....   | 103        |
| 7        | <b>Resumen y Perspectivas .....</b>   | <b>106</b> |
| Anexo A. | <b>Los conceptos de la Concha de Administración .....</b>   | <b>109</b> |
| 1.       | ..... general .....   | 109        |
| 2.       | fuentes y documentos pertinentes .....  | 109        |
| 3.       | Conceptos básicos para Industrie 4.0 .....  | 109        |
| 4.       | El concepto de propiedades .....  | 110        |
| 5.       | El concepto de submodelos .....   | 111        |
| 6.       | Estructura básica del Shell de administración de activos .....  | 112        |
| 7.       | Requisitos .....  | 114        |
| Anexo B. | <b>Las plantillas para las Tablas UML .....</b>   | <b>121</b> |
| Anexo C. | <b>Leyenda para modelado UML .....</b>  | <b>122</b> |
| Anexo D. | <b>UML Metamodel con atributos heredados .....</b>  | <b>124</b> |
| Anexo E. | <b>esquemas XML y completo ejemplo .....</b>  | <b>127</b> |
| 1.       | Los esquemas XML para Shell Administración .....  | 127        |
| 2.       | Esquema para Shell Administración general .....   | 127        |
| 3.       | AAS IEC61360 Tipo de datos .....  | 133        |
| 4.       | Ejemplo XML .....   | 134        |
| Anexo F. | <b>esquema JSON y ejemplos completos .....</b>  | <b>137</b> |
| 1.       | JSON esquema para Shell Administración .....  | 137        |

|          |                    |     |
|----------|--------------------|-----|
| 2.       | Ejemplo JSON ..... | 144 |
| Anexo G. | Bibliografía ..... | 146 |

## Tabla de las Tablas

|  |     |
|--|-----|
| Tabla fases y funciones de tipo e instancia de ciclo 1 Vida .....  | 23  |
| Tabla 2 identificables, atributos y permitió identificadores .....   | 28  |
| Tabla 3 Estructura propuesta para URIs .....   | 30  |
| Tabla 4 Ejemplo URN e identificadores basados en URL de la Shell Administración .....  | 30  |
| Tabla 5 tipos básicos utilizados en Metamodel .....  | 59  |
| Tabla 6 Distinción de diferente formato de datos para el AAS .....   | 63  |
| Tabla 7 Minimal XML para la estructura de nivel superior .....   | 67  |
| Tabla 8 Uso de grupos de modelos XSD .....   | 68  |
| Tabla 9 JSON Minimal para la estructura de nivel superior .....  | 73  |
| Tabla 10 JSON mínimo ejemplar para Referencias .....   | 74  |
| Tabla 11 JSON mínimo Ejemplar para la estructura de nivel superior .....   | 74  |
| Tabla 12 ReferenceElement ejemplar en JSON .....   | 75  |
| Tabla 13 Ejemplar GlobalReference en JSON .....  | 76  |
| Tabla 14 tipo ejemplar de referencia en JSON .....   | 77  |
| Tabla 15 Ejemplo de filtrado de la información en XML .....  | 83  |
| Tabla 16 Conjunto de políticas posibles en función de cómo se firman archivos de paquetes, cómo habilitar una determinada política y las consecuencias de una política ..... | 102 |
| Tabla 17 esquema JSON .....  | 137 |
| Tabla 18 JSON ejemplo completo .....   | 144 |

## Tabla de Figuras

|  |    |
|--|----|
| Figura 1 Uso de Cambio Expediente entre los socios de la cadena de valor .....   | 19 |
| Figura 2 Intercambio de archivos entre dos socios de la cadena de valor .....  | 20 |
| Figura 3 Los tipos ejemplares y los casos de activos representados por múltiples AAS .....   | 24 |
| Figura 4 relaciones ejemplares entre metamodelo de AAS, tipos de AAS y casos AAS .....   | 26 |
| Figura 5 El Shell Administración necesita un identificador único, así como cada uno de los activos que se describe. figura Modificado de [4] ..... | 27 |
| Figura 6 La motivación de identificadores ejemplares y idShort .....   | 32 |
| Figura 7 Descripción general Metamodel de la ..... Shell Administración de Activos .....   | 33 |
| Figura 8 vista general paquete Metamodel .....   | 34 |
| Figura 9 Metamodel para identificables y Referables .....  | 36 |
| Figura 10 Metamodel para Identificador .....   | 38 |
| Figura 11 Metamodel para HasKind .....   | 39 |

|  |                 |
|--|-----------------|
| Figura 12 Metamodel para información administrativa .....  | 40              |
| Figura 13 Metamodel Para referencias semánticas (hasSemantics) .....   | 41              |
| Figura 14 Metamodel Qualificables y limitaciones .....   | 41              |
| Figura 15 Ejemplo Fórmula .....  | 42              |
| Figura 16 Metamodel para HasDataSpecification .....  | 43              |
| Figura 17 Metamodel AssetAdministrationShell .....   | 44              |
| Figura 18 Metamodel de Activos .....   | 45              |
| Figura 19 Metamodel para Submodelo .....   | 46              |
| Figura 20 Metamodel para Tipos Submodelo Element .....   | 48              |
| Figura 21 Metamodel para elementos de datos y sus subtipos .....   | 49              |
| Figura 22 Metamodel para Submodelo Element Colecciones .....   | 52              |
| Figura 23 Metamodel de relación Elementos .....  | 53              |
| Figura 24 Metamodel de Operaciones .....   | 53              |
| Figura 25 Metamodel de Vistas .....  | 54              |
| Figura 26 Metamodel de diccionario de conceptos .....  | 55              |
| Figura 27 Metamodel referencias y Keys .....   | 56              |
| La Figura 28 incorporado en los tipos de Definición de Esquema XML 1.1 (XSD) .....   | 58              |
| Figura 29 Concepto de plantillas de especificación de datos .....  | 60              |
| Figura 30 Especificación de la plantilla de datos para definir Descripciones de las propiedades conformes a la norma IEC 61360 ..... | 60              |
| Figura 31 Vista general descripciones de concepto y plantillas de especificación de datos .....                                      | 61              |
| Figura 32 Vista gráfica en el intercambio de datos Formatos para la Administración de Activos de Shell .....                         | 63              |
| Figura 33 Ejemplo velocidad de rotación para la serialización de los formatos de datos .....   | sesenta y cinco |
| Figura 34 estructura de nivel superior de un entorno AssetAdministration Shell asigna a Esquema XML .....                            | 67              |
| Figura 35 XSD Modelo Grupos .....  | 68              |
| Figura 36 Teclas y Referencias .....   | 69              |
| Figura 37 Vista general en la cartografía y meta-datos .....   | 70              |
| Figura 38 Descripción Concept en XML en general .....  | 70              |
| Figura 39 especificación de datos a través de la norma IEC 61360 propiedad atribuye .....  | 71              |
| Figura 40 estructura de nivel superior de un entorno AssetAdministration Shell asignada a JSON .....                                 | 73 ..           |
| Figura 41 referencia Submodelo en AssetAdministrationShell para JSON .....   | 73              |
| Figura 42 Uso de ReferenceElement en JSON .....  | 75              |
| Figura 43 Uso de GlobalReference en JSON .....   | 76              |
| Figura 44 Tipo de referencia ejemplar en JSON .....  | 77              |
| Figura 45 Ejemplo de filtrado de exportación e importación .....   | 82              |
| Figura 46 Atributo control de acceso basado [22] .....   | 84              |
| Figura 47 Metamodel Descripción general del control de acceso de AAS .....   | 85              |
| Figura 48 Paquetes de Introducción a la seguridad .....  | 86              |
| Figura 49 Metamodel de atributos de seguridad de AAS .....   | 87              |



|  |     |
|--|-----|
| Figura 50 Metamodel para Control de acceso .....   | 87  |
| Figura 51 Metamodel para Control de acceso .....   | 89  |
| Figura 52 Metamodel para la Regla permiso de acceso .....  | 91  |
| Figura 53 modelo lógico para el formato AASX .....   | 97  |
| Figura 54 modelo físico para un AASX basado en un producto de muestra (izquierda) y un ejemplo de mapeo para el modelo lógico (derecha) .....    | 100 |
| Figura 55 conceptos importantes de Industrie 4,0 unidos al activo [2, 23]. 14.0 Componente a ser formado por Shell Administración y activo. .... | 110 |
| Figura 56 Ejemplar definición de una propiedad en el IEC CDD .....   | 111 |
| Figura 57 Ejemplos de diferentes dominios de proporcionar propiedades de submodelos de la Shell Administración .....                             | 112 |
| Figura 58 Estructura básica de la Shell AssetAdministration .....  | 113 |
| Figura 59 Agregación en Metamodel en UML - Leyenda .....   | 122 |
| Figura 60 Asociación en Metamodel en UML - Leyenda .....   | 122 |
| Figura 61 Composición en Metamodel en UML - Leyenda .....  | 122 |
| Figura 62 Identificación en Metamodel en UML - Leyenda .....   | 122 |
| Figura 63 Clases herencia en Metamodel en UML - Leyenda .....  | 123 |
| Figura 64 Inheritance enumeraciones en Metamodel en UML - Leyenda .....  | 123 |
| Figura 65 Modelo Core con atributos heredados .....  | 124 |
| La Figura 66 de control de acceso con atributos heredados .....  | 125 |
| Figura 67 Submodelo Collection Element con la herencia .....   | 126 |

## 1 Preámbulo

## 1.1 notas editoriales

Este documento fue producido septiembre 2017 a julio 2018 por un grupo de trabajo conjunto con miembros de ZVEI SG 'modelos y normas' y Plataforma Industrie 4.0 WG1 grupo de trabajo. El documento fue validado posteriormente por GT1 de la plataforma.

Para una mejor legibilidad, en términos compuestos de la abreviatura "I4.0" se utiliza constantemente para "Industrie 4.0". Se utiliza por sí solo "Industrie 4.0" se sigue utilizando.

## 1.2 Alcance de este documento

El objetivo de este documento es hacer especificaciones seleccionadas de la estructura de la Administración de Shell de tal manera que la información sobre los activos y componentes I4.0 se puede cambiar de manera significativa entre los socios de una red de creación de valor.

Esta parte de este documento, por tanto, se centra en la cuestión de cómo debe ser procesada y estructurado dicha información. Con el fin de hacer que estas especificaciones, el documento estipula formalmente algunos principios estructurales de la Administración de Shell. Esta parte no describe las interfaces técnicas de la Concha de Administración u otros sistemas para el intercambio de información, protocolos o los patrones de interacción.

El documento se refiere a la norma DIN 92000 SPEC proceso de estandarización en paralelo para las declaraciones de valor de la propiedad [15] y refleja aspectos importantes en este documento. Este documento se centra en:

- El transporte de información de un socio en la cadena de valor a la siguiente
- Shell administración, submodelos y sus estructuras
- identificadores
- El acceso concepto de derechos y funciones

Este documento actualmente cuenta con la versión V1.0. Se dirige a ser adecuadamente completo y coherente para ser utilizado como base para desarrollos y como entrada para la discusión con las organizaciones internacionales de normalización y otras cooperaciones. A V1.1 versión se pretende que incluya asignaciones y características adicionales y debe incorporar el aporte de bancos de pruebas de validación y estandarización internacional.

Las definiciones de la forma y del documento deben ser tales que los departamentos de desarrollo de las redes de creación de valor tienen suficiente información detallada para empezar a trabajar en los sistemas internos para el intercambio de información y de bases de datos correspondientes.

## 1.3 Estructura del documento

Capítulo 2 se resumen, el contenido relevante existente en la estandarización de Industrie 4.0. En otras palabras, esta cláusula proporciona una visión general y explica los motivos, y no es absolutamente necesaria para la comprensión de las definiciones siguientes.

Capítulo 3 se establece suficientes principios estructurales de la Concha de Administración de una manera formal con el fin de garantizar un intercambio de información entre las cáscaras de la Administración. Un extracto de un diagrama de UML está redactada para este propósito. Una discusión más amplia UML que no establece normas se puede encontrar en el anexo. Capítulo 4 proporciona definiciones detalladas para el intercambio de información I4.0 conforme en formatos de datos existentes, como XML, AutomationML, modelos de información OPC UA, JSON o RDF. Se proporciona una explicación para cada uno de estos formatos de datos indicando cómo la información va a ser representado (metamodelo), y se proporciona un ejemplo de una representación. Capítulo 5, 6, 7 describe la promoción de modelos de acceso en función de los atributos de seguridad de la información. Capítulo 9 describe,

## Principios de la obra

## 1.4

El trabajo se basa en el siguiente principio: lo más simple posible, sólo se describen las cosas absolutamente necesarias.

Para la creación de una especificación detallada de la Shell Administración de acuerdo con el alcance de la parte 1 (• 1.2), los papeles de Resultados publicados por Plattform Industrie 4.0, se analizaron la cooperación trilateral con Francia e Italia y los resultados internacionales de normalización y toma como fuente de requisitos para el proceso de especificación. Como se consideraron todas las ideas posibles de los documentos de debate.

Los socios representados en la Plataforma Industrie 4.0 y asociaciones como la ZVEI, la VDMA, VDI / VDE y Bitkom, aseguran que hay una amplia cobertura sectorial, tanto en el proceso, híbrido y automatización de fábrica y en la integración de tecnología de la información (IT) y la tecnología operativa (OT).

alternativas de diseño se discutieron intensamente dentro del grupo de trabajo. Un proceso de retroalimentación extensa con la denominada "caja de resonancia" de esta serie de documentos, con los grupos de trabajo de la Plattform y con los socios asociados se dedicaban sobre las alternativas de diseño y el contenido final de la especificación.

principio rector de la especificación era proporcionar información detallada, que puede ser implementado fácilmente también por pequeñas y medianas empresas.

## 1.5 Términos y Definiciones

aviso hacia adelante

Definición de términos sólo son válidos en un contexto determinado. El glosario actual está relacionada con el contexto de este documento.

control de acceso

protección de los recursos del sistema contra el acceso no autorizado; un proceso por el cual se regula el uso de los recursos del sistema de acuerdo con una política de seguridad y se permite solamente por entidades autorizadas (usuarios, programas, procesos, u otros sistemas) de acuerdo con que la política

- [FUENTE: IEC 62443-1-1 TS]

solicitud

software elemento funcional específica a la solución de un problema de medición y control de procesos industriales

Nota: Una aplicación puede ser distribuida entre los recursos y puede comunicarse con otras aplicaciones.

- [FUENTE: IEC TR 62390: 2005-01, 3.1.2]

activo

objeto físico o lógico propiedad de o bajo las tareas de custodia de una organización, tiene o bien un valor percibido o real en la organización

Nota: En el caso de los sistemas de automatización y control industrial, los activos físicos que tienen el mayor valor directamente medible puede ser el equipo bajo control.

- [FUENTE: IEC 62443-1-1 TS: 2009, 3.2.6]

#### shell administración activo (AAS)

representación digital estandarizada del activo, piedra angular de la interoperabilidad entre las aplicaciones de gestión de los sistemas de fabricación. Identifica el Shell de administración y los activos representados por él, tiene modelos digitales de diversos aspectos (submodelos) y describe la funcionalidad técnica expuesta por el Shell de administración o de los activos respectivos.

**Nota:** Los activos de Shell Shell Administración y Administración son utilizar como sinónimos.

- [SOURCE: Glosario Industrie 4.0]

#### atributo

elemento de datos para la descripción por ordenador sensible de una propiedad, una relación o una clase

- [Fuente: Guía ISO / IEC 77-2]

#### clase

Descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y semántica

- [FUENTE: IEC TR 62390: 2005-01, 3.1.4]

#### componente

producto utilizado como un constituyente en un producto, sistema o planta de ensamblado

- [FUENTE: IEC 61666: 2010, 3.6]

#### concepto

unidad del conocimiento creado por una combinación única de características

- [FUENTE: IEC 61360-1, ISO 22274: 2013, 3.7]

#### identificador (ID)

información de identidad que distingue inequívocamente una entidad de otra en un dominio dado

**Nota:** Hay identificadores específicos, por ejemplo, identificador único UUID universal, IEC 15418 (GS1).

- [SOURCE: Glosario Industrie 4.0]

#### ejemplo

hormigón componente, claramente identificable de un cierto tipo

**Nota:** Se convierte en una entidad individual de un tipo, por ejemplo un dispositivo, mediante la definición de los valores de una propiedad. **Nota:** En una vista orientada a objetos, un ejemplo denota un objeto de una clase (de un tipo).

## 14 | PREÁMBULO

- [SOURCE: IEC 62890: 2016, 3.1.16] 65/617 / CDV

### operación

#### realización de una función ejecutable

Nota: El término método es sinónimo de operación

Nota: una operación tiene un nombre y una lista de parámetros [ISO 19119: 2005, 4.1.3]

- [SOURCE: Glosario Industrie 4,0 (trabajo en progreso)]

### propiedad

#### característica definida adecuada para la descripción y la diferenciación de productos o componentes

Nota: El concepto de tipo e instancia se aplica a las propiedades.

Nota: Esta definición se aplica a propiedades tales como se describe en la norma IEC 61360 / ISO 13584-42 Nota: Los tipos de propiedades se definen en los diccionarios (como componente de diccionario de datos IEC o eCI @ ss), no lo hacen

tener un valor. El tipo de propiedad es también llamado tipo de elemento de datos en algunas normas. Nota: Los casos de propiedad tienen un valor y proporcionados por los fabricantes. Un ejemplo es la propiedad también par llamada propiedad-valor en ciertas normas.

Nota: Las propiedades incluyen valor nominal, valor real, las variables de tiempo de ejecución, los valores de medición, etc. Nota: Una propiedad describe una característica de un objeto dado. Nota: Una propiedad puede tener atributos tales como el código, la versión y revisión.

Nota: La especificación de una propiedad puede incluir opciones predefinidas de valores.

- [SOURCE: según la Guía ISO / IEC 77-2], así como [SOURCE: según Glosario Industrie 4.0]

### índice

elemento asociado con una instancia de la propiedad o elemento submodelo bien definidos, la restricción de la declaración de valor para un determinado período de tiempo o caso de uso

Nota: calificador puede tener un valor asociado

- [SOURCE: según IEC 62569-1]

### variable

entidad de software que pueden tomar valores diferentes, uno a la vez

- [FUENTE: IEC 61499-1]

### ver

proyección de un modelo o modelos, que se ve desde un punto de vista perspectiva o dado y omite las entidades que no son relevantes para esta perspectiva

- [FUENTE: lenguaje de modelado unificado - UML]

### representación virtual

totalidad de la información de la Shell Administración, tales como submodelos, propiedades y objetos de datos complejas, que cubren los modelos digitales para el activo respectivo (s) para todas las fases del ciclo de vida aplicables

- [Fuente: [18]]

de fabricación inteligentes

enfoque de fabricación, que mejora sus aspectos de rendimiento con el uso integrado e inteligente de los procesos y recursos en la lucha virtual, física y esferas humanas para crear y entregar productos y servicios, que también colabora con otros dominios dentro de cadenas de valor de una empresa.

Nota: Los aspectos de rendimiento incluyen la agilidad, la eficiencia, la seguridad, la sostenibilidad o cualquier otra actuación indicadores identificados por la empresa. Nota: Además de la fabricación, otros dominios empresariales pueden incluir ingeniería, logística, marketing, compras, ventas o cualquier otro dominios identificados por la empresa. Nota: esta definición es, en noviembre de 2018, en discusión dentro del grupo de trabajo conjunto ISO / IEC (GMT) 21. Sin embargo, se da una buena indicación y una fuente citable.

- [SOURCE: ISO / TMB / SMCC]

submodelo

utilizado para estructurar la representación virtual y funcionalidad técnica de una cáscara de Administración en partes distinguibles. Cada submodelo refiere a un dominio bien definido o materia. Submodelos pueden convertirse en tipos submodelos estandarizados y por lo tanto convertirse. Submodelos pueden tener diferentes ciclos de vida.

Nota: El concepto de tipo e instancia se aplica a submodelos.

elemento submodelo

elemento adecuado para la descripción y la diferenciación de los activos

Nota: se extiende la definición de propiedades Nota: podría comprender operaciones, objetos binarios

sistema

interactuando, relacionados entre sí, o elementos interdependientes que forman un todo complejo

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

funcionalidad técnica

la funcionalidad de la Concha de Administración que se expone mediante una interfaz de programación de aplicaciones (API) y que es la creación de valor añadido a los activos (s) respectivos.

- Nota: puede consistir en elementos individuales, que también se conocen como funciones, operaciones, métodos, habilidades.
- [SOURCE: según [18]]

modelo

especificación de las características comunes de un objeto en suficiente detalle para que dicho objeto puede ser instanciada usarlo

Nota: objeto puede ser cualquier cosa que tiene un tipo de

- [SOURCE: según la norma ISO / IEC 10746-2]

tipo

hardware o software elemento que especifica los atributos comunes compartidos por todas las instancias del tipo de

- [FUENTE: IEC TR 62390: 2005-01, 3.1.25]



## 1.6 abreviaturas

| Abreviatura           | Descripción  |
|-----------------------|--|
| AAS                   | Shell de administración de activos   |
| AASX                  | extensión de formato de archivo de paquete para el AAS                               |
| AML                   | AutomationML   |
| API                   | Los programadores de aplicaciones de interfaz  |
| BITKOM                | Bundesverband Informationswirtschaft, Telekommunikation und neue Medien<br>mi. V.    |
| GOTA                  | Binary Large Object  |
| CDD                   | Diccionario de datos común   |
| GUID                  | identificador único global   |
| I4.0                  | Industrie 4.0  |
| CARNÉ DE IDENTIDAD    | identificador  |
| IEC                   | Comisión Electrotécnica Internacional  |
| IRDI                  | Identificador Internacional de Datos de Registro                                     |
| YO ASI                | Organización Internacional para la Estandarización                                   |
| JSON                  | JavaScript Object Notation   |
| MÍMICA                | Multipurpose Internet Mail Extensions  |
| OPC UA                | Arquitectura unificada para el Object Linking and Embedding for Process Control      |
| PDF                   | Formato de Documento Portable  |
| RAMI4.0               | Arquitectura de referencia Modelo Industrie 4.0                                      |
| RDF                   | Resource Description Framework   |
| DESCANSO              | Transferencia de estado representacional   |
| RFC                   | Petición de comentario   |
| ROA                   | Ressource Oriented Architecture  |
| SOA                   | Arquitectura orientada a Servicios   |
| PASO                  | Estándar para el intercambio de datos de modelo del producto                         |
| UML                   | Lenguaje de Modelado Unificado   |
| URI, URL, URN Uniform | Resource Identifier, localizador, Nombre VDE   |
|                       | Verein Deutscher Ingenieure  |
| VDE                   | Verband der Elektrotechnik Elektronik Informationstechnik correo. V.                 |
| VDMA                  | Verband Deutscher Maschinen und Anlagenbau eV  |
| W3C                   | Consorcio Mundial de la red  |
| XML                   | eXtensible Markup Language   |
| cremallera            | formato de archivo de almacenamiento que soporta la compresión de datos sin pérdidas |
| ZVEI                  | Zentralverband Elektrotechnik- und Elektronikindustrie correo. V.                    |

## 2 Conceptos básicos y la imagen que lleva

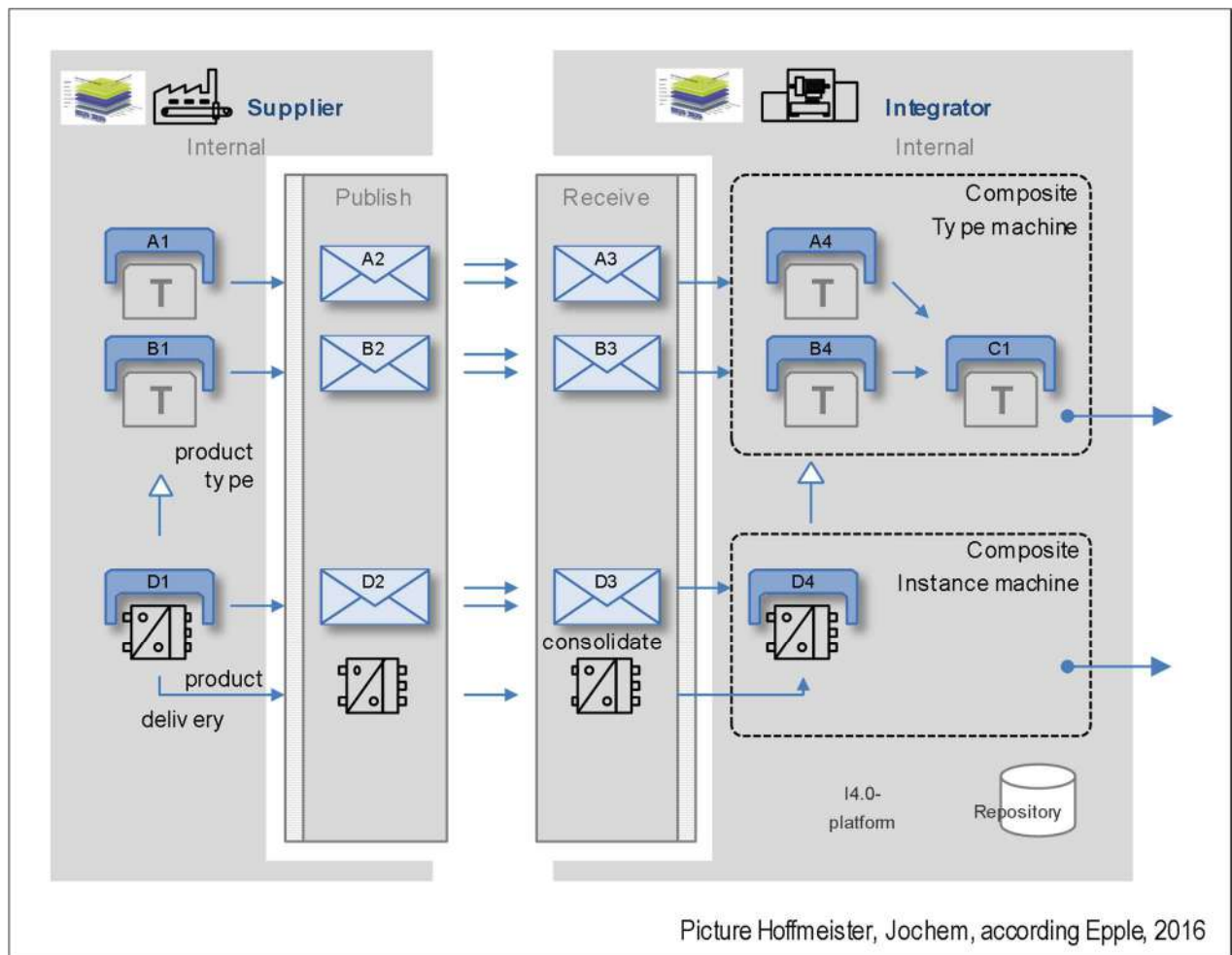
## 2.1 Conceptos básicos

Muchos conceptos para Industrie 4.0 y fabricación inteligente ya existen. Los más importantes se resumen en el anexo informativo A.

## 2.2 foto conduce

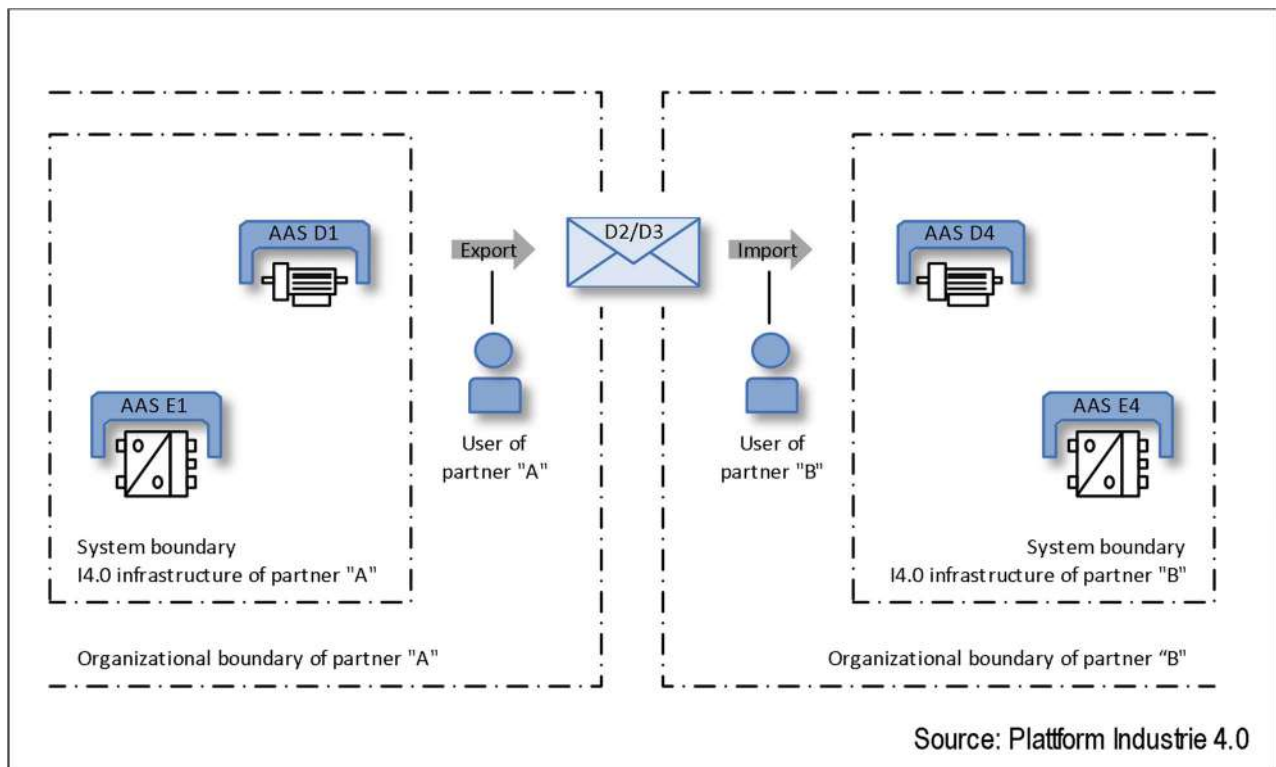
El caso que lleva su uso en este documento es el intercambio de un Shell de administración de activos, incluyendo todos sus documentos auxiliares y artefactos de socio de la cadena un valor a otro. Esto es, que en este documento no se ocupan de los casos de uso de los ya desplegados cáscaras de administración de activos que se ejecutan en una infraestructura específica, pero sólo con el intercambio de archivos entre los socios.

Figura 1 Uso de Cambio Expediente entre los socios de la cadena de valor



La Figura 1 muestra el cuadro general. Representa dos socios de la cadena de valor; "Proveedor" va a ofrecer algunos productos, "integrador" va a utilizar estos productos con el fin de construir una máquina. se proporcionan dos tipos de conchas de la Administración; uno para el activo siendo el tipo de un producto, una para los activos que las instancias real del producto. "Proveedor" y "integrador" se están formando dos cuerpos legales independientes (Figura 2).

Figura Cambio 2 de archivos entre dos socios de la cadena de valor



El intercambio de archivos tiene que cumplir con algunos requisitos con respecto a la facilidad de uso y seguridad. Es necesario que exista un acuerdo bilateral sobre las restricciones de seguridad que deben cumplirse para la transferencia y el uso de los archivos. Esto se explica con más detalle en el capítulo 5.

Para facilidad de uso de un formato de contenedor para el intercambio de archivos se utiliza y se define una estructura correspondiente (véase la cláusula 6). Esta estructura predefinida ayuda al consumidor a comprender el contenido de los archivos individuales. Esto es importante porque una especificación AssetAdministration Shell se puede transmitir a través de varios archivos. Adicionalmente, el recipiente puede contener archivos referencias auxiliares por el AAS o incluso código ejecutable.



### 3 El metamodelo de la Concha de Administración

### 3.1 Introducción

Esta cláusula especifica el metamodelo de la información de la Concha AssetAdministration. Antes de hacerlo algún aspecto general de la manipulación de los tipos de activos y se describen casos (véase la cláusula 3.2 Tipos e instancias). Otro aspecto muy importante de la AAS es el aspecto de identificación, véase la cláusula 3.3 Identificación de las entidades. El metamodelo para aspectos de seguridad del Shell de administración se describe en la cláusula 5.

La leyenda para la comprensión de los diagramas de UML y la especificación de la tabla de las clases se encuentran en el anexo B y el anexo C.

### 3.2 Tipos e instancias

#### 3.2.1 Ciclo vital

Industrie 4.0 utiliza una comprensión ampliada de activo, que comprende elementos tales como fábricas, sistemas de producción, equipos, máquinas, componentes, productos elaborados y materias primas, procesos de negocio y las órdenes, activos inmateriales (como los procedimientos, software, documentos, planes, propiedad intelectual, normas), servicios y personal humano y más.

El modelo RAMI4.0 [3] cuenta con uno, eje del ciclo de vida generalizada, que se deriva de la norma IEC 62890. La idea básica es distinguir de todos los activos en Industrie 4.0 entre los posibles tipos y ejemplo. Esto hace que sea posible la aplicación de la distinción tipo / instancia para todos los elementos tales como el tipo de material / ejemplo de material, tipo de producto / instancia de producto, tipo de máquina / instancia de la máquina y más. información relacionada negocios será manejado en la capa 'Business' del modelo RAMI4.0, así, que abarca también detalles de la orden y flujos de trabajo, de nuevo con los tipos / instancias.

Tabla fases y funciones de tipo e instancia de ciclo 1 Vida

| Fase    |                     | Descripción   |
|---------|---------------------|---|
| Tipo    | Desarrollo          | Válido desde el ideación / conceptualización a la primera prototipos / prueba. El 'tipo' de un activo se define, y las propiedades distintivas y funcionalidades se definen e implementan. Todos los artefactos de diseño (internos) se crean, como los datos de CAD, esquemas, software embebido, y se asocian con el tipo de activos.   |
|         | Uso / Mantenimiento | El aumento gradual de la capacidad de producción. Se crea la información 'externa' asociado al activo, tales como hojas de datos técnicos, información de marketing. El proceso de venta comienza.  |
| Ejemplo | Producción          | casos activos se crean / producen, sobre la base de la información de tipo de activos. La información específica acerca de la producción, logística, calificación y prueba están asociados con las instancias de activos.   |
|         | Uso / Mantenimiento | Fase de utilización por parte del comprador de los casos activos. Los datos de uso se asocia con la instancia de activos y podría ser compartida con otros socios de la cadena de valor, como el fabricante de la instancia de activos. También se incluye: mantenimiento, re-diseño, la optimización y la clausura de la instancia de activos. La historia completa del ciclo de vida se asocia con el activo y podría ser archivado / compartido para la documentación. |

La tabla 1 da una visión general de las diferentes fases ciclos de vida y el papel de tipo e instancia en estas fases: La relación más importante es entre tipos de activos y ejemplo de activos. Esta relación debe mantenerse durante toda la vida de las instancias de activos. Por esta relación, cambios a los tipos de activos pueden ser enviados a los casos activos, ya sea de forma automática o bajo demanda.

**Nota:** para la distinción de 'tipo' y 'ejemplo', el término 'clase' se utiliza en el presente documento.

La segunda clase de relaciones son bucles / información dentro del ciclo de vida del tipo de activos y la instancia de retroalimentación. Para los activos de productos, por ejemplo, información sobre el uso y el mantenimiento de instancias de productos puede mejorar la fabricación de productos como también causan mejoras de diseño para la (siguiente) tipo de producto.

La tercera clase de relaciones son de intercambio / información de prealimentación con activos de otras clases de activos. Por ejemplo, el abastecimiento a la información de activos de la empresa puede influir en aspectos de diseño de productos; o, el diseño de los productos afecta el diseño de la línea de fabricación.

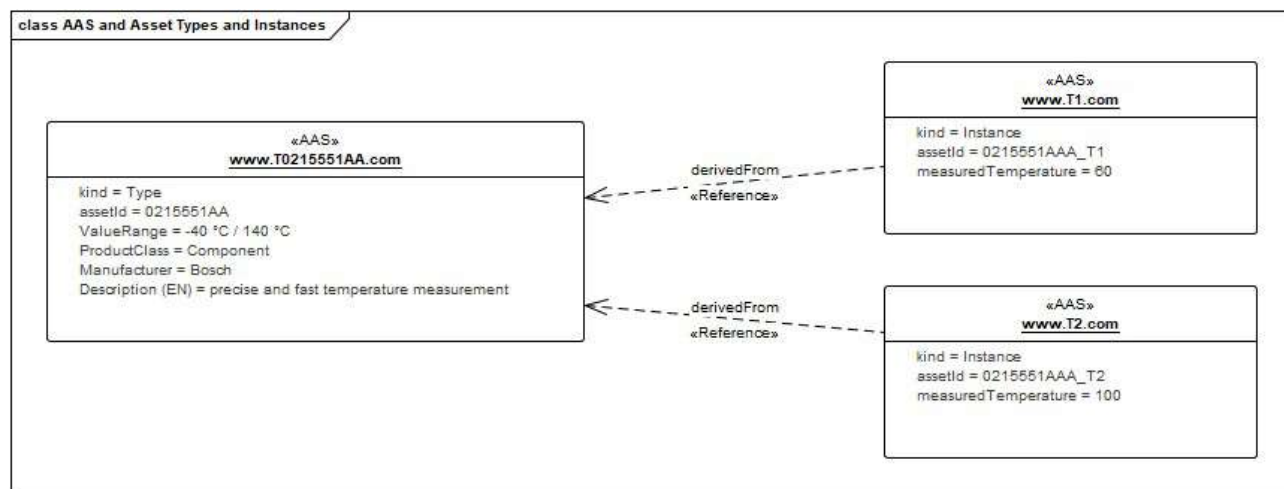
**Nota:** Para ver una ilustración de la segunda tercera clase / de relaciones confiere el modelo del NIST, también.

Una clase adelante de las relaciones son entre activo de diferentes niveles de jerarquía. Por ejemplo, éstas podrían ser las relaciones (dinámico) entre las estaciones de fabricación y productos que se producen actualmente. Estos podrían ser también las descomposiciones de los sistemas de producción en las jerarquías físicas, funcionales o de seguridad. Por esta clase de relaciones, equipos de automatización se explica como un complejo gráfico, interrelacionado de dispositivos y productos de automatización, la realización de las tareas de producción y el autoaprendizaje / inteligentes de optimización.

### 3.2.2 Ejemplo

La siguiente figura muestra un ejemplo para el manejo de los tipos de activos y las instancias de activos, la manipulación alguna información a modo de ejemplo también. Una explicación más detallada seguirá los siguientes cláusulas.

Figura 3 Los tipos ejemplares y los casos de activos representados por múltiples AAS



**Nota:** El ejemplo se ha simplificado para facilitar la comprensión y no sólo se cumple aproximadamente con el metamodelo, ya que es específica en la cláusula 4. El manejo de id se simplifica así: los nombres de las clases corresponden a la identificador global único de la EAA. **Nota:**

En el contexto de la plataforma Industrie 4.0 tipos e instancias típicamente se refieren a "tipos de activos" y "casos activos". Cuando se hace referencia a los tipos o instancias de un AAS Esto se indica explícitamente como "tipos" y "AAS AAS casos" para no mezclar ambos.

**Nota:** Por favor refiérase a la cláusula 1.5 para la definición de tipos de IEC e instancias. Para el alcance de este documento, hay equivalencia total entre estas definiciones y los conceptos tipo / instancia de programación orientada a objetos (OO).

Habrán un tipo de activo concreto de un sensor de temperatura y dos sensores de temperatura físicas único de identificación de este tipo. La intención es proporcionar un AAS separado para el tipo de recurso, así como para cada instancia activo individual. En el ejemplo el primer sensor tiene el ID único "0215551AA\_T1" y el segundo sensor tiene el ID único "0215551AA\_T2". El AAS para el primer sensor tiene la URL única "www.T1.com" y el AAS para el segundo sensor tiene la URL única "www.T2.com". El tipo para ambos es "Instancia". El ejemplo muestra que la temperatura medida en el tiempo de funcionamiento de los dos sensores es diferente: para T1 es 60 ° C, para T2 es 100 ° C. Para el bienestar tiempo ignoramos la relación "derivedFrom" de las dos AAS "T1" y "T2" con AAS "www.T0215551AA.com".



Nota: La unidad se puede obtener por la referencia semántica del elemento "measuredTemperature". Por simplicidad esto no se muestra en el ejemplo.

Estos dos casos activos tienen una gran cantidad de información que comparten: la información del tipo de activo (en este ejemplo, un tipo de sensor). Para este tipo de activos se crea una propia AAS. El ID único para este AAS es "www.T0215551AA.com", el ID único del tipo de sensor es "0215551AA". El tipo en este caso es "Tipo" y no "Instancia". La información que es la misma para todas las instancias de este tipo de sensor de temperatura es la ProductClass ("Componente"), el fabricante ("Bosch") y la Descripción Inglés "precisa y la medición de temperatura rápido", así como la rango de valores "-40 ° C / 140 ° C".

Ahora las dos AAS de las dos instancias de activos pueden referirse a la AAS del tipo de activo "0215551AA" mediante el atributo de relación "derivedFrom".

Nota: "Atributo" se refiere en el sentido de UML a la propiedad o característica de una clase (instancia). Nota: Normalmente, si existe un tipo activo específico, que existe en el tiempo antes de que los casos de activos respectivos. Nota: Un AAS se utiliza sinónimo de una instancia AAS. Un AAS se puede realizar sobre la base de un tipo de AAS. AAS son tipos

fuera del alcance de este documento. Nota:

En la normalización pública los tipos de AAS pueden ser estandarizados. Sin embargo, es mucho más importante estandarizar los tipos de propiedad (definiciones o descripciones de propiedad llamado concepto) o cualquier otro elemento submodelo mecanografiadas, así como los tipos de submodelos completos porque aquellos pueden ser reutilizados en diferentes AAS. Nota:

En el dominio de Internet de los objetos (IO), los casos de activos suelen ser denotados como "cosas" mientras que los tipos de activos se indican como "producto".

### 3.2.3 Metamodelo de los tipos e instancias de Shell Administración de Activos

En el anterior tipo de cláusula y se explicaron ejemplos de activos. Obviamente, la pregunta entonces surge cómo armonizar AAS, así como los tipos de AAS. En nuestro ejemplo se puede observar que los atributos "assetId" y "tipo", así como el identificador global (id, representado como el nombre de la clase) están presentes para todos los AAS. Sin embargo, si no existe una norma, no está claro que la semántica de "assetId" y "clase" "id" son los mismos y no está claro cuál de los atributos son obligatorios y cuáles son específicos para el activo ( tipo o ejemplo). Esto se ilustra en la Figura 4. Esta es la tarea de este documento: La definición de un metamodelo que define qué atributos son obligatorios y que son opcionales para todos AAS. La Plataforma Industrie 4,0 metamodelo Para los depósitos AssetAdministration se define en la cláusula 3.

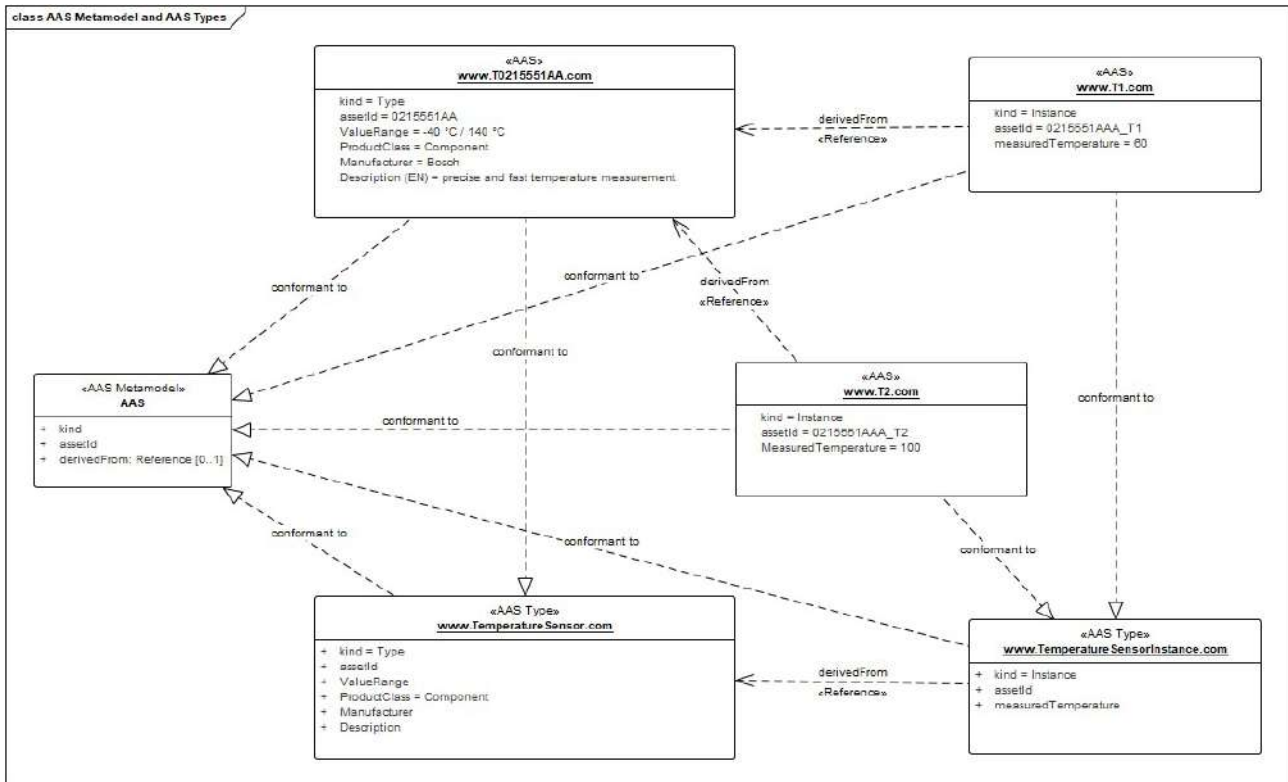
Nota: Este enfoque garantiza que el requisito se cumple tAAS- # 19. Otro enfoque podría haber sido definir dos metamodelos: una para los tipos de activos y uno para los casos de activos. Sin embargo, el gran conjunto de similitudes motivó a ir con uno metamodelo.

Nota: El metamodelo en sí no prescribe submodelos obligatorios. Este es otro paso de la normalización similares a la prescripción de submodelos de nivel de tipo de AAS.

Nota: Un tipo de AAS se realizó sobre la base del metamodelo de un AAS como se define en este documento. este metamodelo que se conoce como la "AAS Metamodel" Nota:

No es obligatorio definir un tipo de AAS antes de definir un AAS (ejemplo). Una instancia AAS que no se da cuenta un tipo de AAS se realizó sobre la base del metamodelo de un AAS como se define en este documento.

Figura 4 relaciones ejemplares entre metamodelo de AAS, tipos de AAS y casos AAS



### 3.3 Identificación de las entidades

#### 3.3.1 Visión general

Identificadores son necesarios de acuerdo con [4] para la identificación única de muchas entidades diferentes dentro del dominio de la fabricación inteligente. Por esta razón, son un elemento fundamental de una descripción formal de la Concha de Administración. Especialmente, es al menos requiere una identificación para:

- Los depósitos de administración de activos,
- bienes,
- instancias submodelos y tipos de submodelos,
- las definiciones de propiedades / descripciones de conceptos en los repositorios externos, como eCI @ ss o IEC CDD

La identificación se llevará a cabo durante dos fines:

- (1) para distinguir de forma única a todas las entidades de una cáscara de Administración, y
- (2) relacionar entidades a las definiciones externas, tales como tipos de submodelos y definiciones de propiedades, a fin de obligar una la semántica a estos datos y las entidades funcionales de una cáscara de Administración.

#### 3.3.2 Lo que existen ifiers Ident?

En [4], [20], se definen dos tipos de identificación globales conforme estándar:-

**(una) IRDI - ISO29002-5, ISO IEC 6523 e ISO IEC 11179-6 [20] como un esquema de identificador para las propiedades y**

clasificaciones. Se crean en un proceso de especificación del consorcio a gota o la normalización internacional. Con este fin, los usuarios se sientan juntos y alimentar a sus ideas en los consorcios u organismos de normalización. Propiedades de ayuda ISO, IEC para salvaguardar los intereses comerciales clave. Repositorios como eCI @ ss y otros hacen posible la estandarización de un número relativamente grande de identificadores en un corto período de tiempo apropiada.

(segundo) URI - URI y URL como la identificación de activos, depósitos de administración y otros (probablemente no estandarizado, pero globalmente único) propiedades y clasificaciones.

También se permite lo siguiente:

(do) Personalizado - Identificadores personalizados internos como GUID (identificadores únicos globales), que un fabricante puede utilizar para todo tipo de propósitos internos dentro de la Administración de Shell.

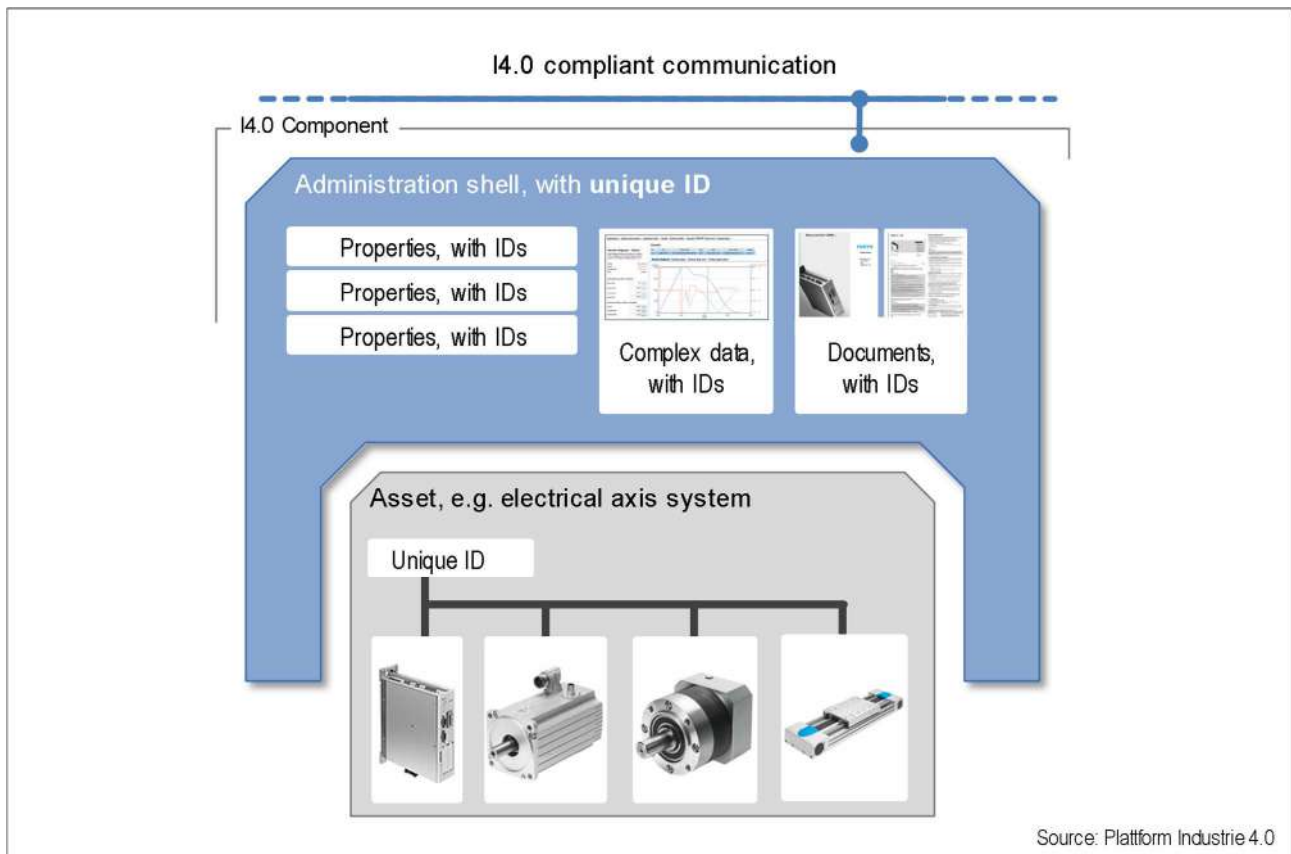
Esto significa que los URI / URL e identificadores personalizados internos pueden representar y comunicar la información específica del fabricante y funciones en el Shell de administración y la infraestructura 4.0 tan bien como la información y funciones estándar. Una infraestructura puede servir a dos propósitos.

Además de los identificadores globales también hay identificadores que son únicas sólo dentro de un espacio de nombres definido, por lo general su elemento padre. Estos identificadores son también llamados identificadores locales. Ejemplo: Propiedades dentro de un submodelo tienen identificadores locales.

### 3.3.3 Identificadores para los activos y las cáscaras de la Administración

Para el dominio de la fabricación inteligente, los activos deben ser identificados única en el mundo [4] [20] por medio de identificadores (IDs). El Shell de administración tiene un identificador único, también.

Figura 5 El Shell Administración necesita un identificador único, así como cada uno de los activos que se describe. figura Modificado de [4]



Un Shell Administración representa exactamente un activo, con un ID activo único. En una producción basada lotes, los lotes se convertirán en el activo y se describirán por un Shell Administración respectiva. Si un conjunto de activos se designarán mediante una cáscara Administración, un identificador único para el activo compuesto tiene que ser creado [12].

El ID del activo debe cumplir las restricciones de los identificadores globales de acuerdo [4] [20]. Si el activo se cuenta con otras identificaciones, números de serie y demás, no hay que confundir con los identificadores únicos globales del propio activo <sup>1</sup>.

### 3.3.4 ¿Qué Identificadores de usar para los cuales las entidades

No cada identificador es aplicable para todas las entidades del modelo UML; Por lo tanto, la siguiente tabla pone restricciones sobre las diversas entidades, que implementan "identificación" o "hasSemantics". Los atributos se relacionan con el metamodelo en la cláusula 3.4.

Tabla 2 identificables, atributos y permitió identificadores

| identificable                  | Atributo           | Identificadores permitidos | observaciones   |
|--------------------------------|--------------------|----------------------------|---|
| Activo<br>AdministrationShell  | carne de identidad | URI                        | obligatoria Típicamente, se utilizarán las URL  |
|                                | idShort            | cuerda                     | n / A   |
| Activo                         | carne de identidad | URI                        | obligatoria Típicamente, se utilizarán las URL [4]  |
|                                | idShort            | cuerda                     | obligatorio   |
| Submodelo con tipo = Tipo      | carne de identidad | IRDI, URI                  | obligatorio<br>IRDI, si el sub-modelo definido ha sido estandarizada y un IRDI se aplicó por ella   |
|                                | idShort            | cuerda                     | obligatorio<br>Normalmente se utiliza como idShort para el submodelo de la especie, así Instancia   |
|                                | semanticId         | IRDI, URI                  | Opcional<br>El ID semántico podría referirse a una fuente de información externa, lo que explica la formulación del submodelo (por ejemplo, un PDF si un estándar)  |
| Submodelo con tipo = Instancia | carne de identidad | URI, personalizada         | obligatorio   |
|                                | idShort            | cuerda                     | obligatorio<br>Típicamente, el IdShort o nombre corto del tipo submodelo referenciados a través de semanticId   |
|                                | semanticId         | IRDI, URI                  | Opcional<br>El tipo submodelo puede ser o bien una referencia a un submodelo con tipo = Tipo (dentro de la misma o de otra AAS) o puede ser una referencia externa a un estándar externo que define la semántica de la submodelo. |
| SubmodelElement                | semanticId         | IRDI, URI, de encargo      | obligatoria (ver Constraint);<br><b>vínculos con la conceptDescription o la definición de concepto en un repositorio externo a través de una identificación global</b>  |
|                                | idShort            | cuerda                     | obligatorio   |

<sup>1</sup> Tales identificadores locales adicionales están contenidos en el submodelo " assetIdentificationModel ".

|                        |                    |                   |   |
|------------------------|--------------------|-------------------|---|
|                        |                    |                   | Normalmente, el nombre corto del elemento de referencia a través de semanticId  |
| ConceptDescription     | carne de identidad | Costumbre o IRDI  | obligatorio<br><br>ConceptDescription tiene que tener una identificación global. Si la descripción del concepto es una copia de un diccionario externo como eCI @ ss puede utilizar el mismo ID global a medida que se utiliza en el diccionario externo. |
|                        | idShort            | cuerda            | n / A<br><br>o mismo nombre que corta   |
|                        | isCaseOf           | IRDI, URI         | Opcional<br><br>enlaces a la definición del concepto en un repositorio externo de la descripción concepto es una copia de, o que corresponde a  |
|                        | semanticId         | n / A             | n / A<br><br>la descripción concepto define la semántica, si se refleja otra definición del concepto en un diccionario externo a continuación, se debe utilizar isCaseOf  |
| Ver<br><br>Calificador | semanticId         | IRDI, URI         | enlaces a la definición de la vista en un repositorio externo   |
|                        | idShort            | cuerda            | obligatorio   |
|                        | semanticId         | IRDI, URI interna | Enlaces a la definición de tipo de clasificación en un repositorio externo  |

### 3.3.5 ¿Cómo se crean nuevos ifiers Ident?

Después de los diferentes tipos de identificación de la cláusula 3.3.3, se puede afirmar:

**(una)** IrDI se supone que ya se existente por un proceso de especificación y estandarización externa, cuando se viene a la creación de un cierto Shell Administración. Para traer dichos identificadores IrDI en la vida, véase la cláusula 4 del documento [4].

**(segundo)** URIs y URLs pueden formarse fácilmente por los desarrolladores de sí mismos, también sobre la marcha cuando se crea una cierta

Shell administración. Todo lo que se necesita es un nombre de host URL válida, por ejemplo de la empresa, y para asegurarse de que la forma en que se organizó el dominio (por ejemplo [www.festo.com](http://www.festo.com)) asegura que el camino detrás del nombre de host está reservado de manera exclusiva semánticamente para estos identificadores. De esta manera, cada desarrollador puede crear un URI o URL arbitraria mediante la combinación del nombre de host y un poco de camino elegido, que sólo tiene que ser único en la organización del desarrollador.

**(do)** identificadores personalizados también pueden formarse fácilmente por los propios desarrolladores. Todo lo que se necesita es una

funcionalidad programática correspondiente <sup>2</sup> para ser recuperado. Es necesario garantizar que Identificadores personalizados internos pueden distinguirse claramente de (a) o (b).

**(re)** Los identificadores locales también pueden ser creados sobre la marcha. Tienen que ser único dentro de su espacio de nombres, por lo general definido por el padre relación.

<sup>2</sup> [https://en.wikipedia.org/wiki/Universally\\_unique\\_identifier](https://en.wikipedia.org/wiki/Universally_unique_identifier)

## 3.3.6 Las mejores prácticas para la creación de identificadores URI

El enfoque para la semántica y la interacción para I4.0 componentes [17] sugiere el uso de la siguiente estructura para URIs<sup>3</sup>, que está ligeramente modificado aquí. Idea es estructurar siempre URI siguiendo un esquema de los diferentes elementos:

Tabla 3 Estructura propuesta para URIs

| Elemento  | Descripción  | componente sintaxis |
|---|--|---------------------|
| Organización  | cuerpo legal, unidad administrativa o compañía emisora de la ID  | UNA                 |
| unidad secundaria Organización /<br>unidad secundaria Documento /<br>Document | entidad Sub en la organización anteriormente, o liberado especificación o publicación de organización anteriormente. | PAG                 |
| Submodelo / domain-id sub-modelo  | de dominio funcional o conocimiento racional de activo o Shell administración, el identificador pertenece.           | PAG                 |
| Versión   | Número de versión en línea con la liberación de la especificación o la publicación de Identificador                  | PAG                 |
| Revisión  | Número de revisión en línea con la liberación de la especificación o la publicación de Identificador                 | PAG                 |
| Propiedad / Elemento-ID   | Inmuebles o más ID de elemento estructural de la Concha de Administración  | PAG                 |
| número de instancia   | numeración individual de los casos dentro de la liberación de la especificación o publicación                        | PAG                 |

En la tabla, componente sintaxis "A" se refiere a la autoridad de RFC 3986 (URI) y el identificador de espacio de nombres de RFC 2141 (URNA); "P" se refiere a la ruta de RFC 3986 (URI) y espacio de nombres cadena específica de RFC 2141 (URN).

Utilizando este esquema, urnas y URLs válidos pueden ser creados, siendo ambos URI. Para el uso de las cáscaras de la Administración, se prefieren las URL, como la funcionalidad (tal como servicios REST) puede estar unido a los identificadores, así. Ejemplos de tales identificadores se dan en la Tabla 4.

Tabla 4 Ejemplo URN y basados en URL identificadores de la Shell Administración

| identificador                      | Descripción                                     | clase de propiedad                                | Ejemplos   |
|------------------------------------|---|---|--|
| Administración Shell ID            | Identificación de la Concha de Administración   | Base  | urn: ZVEI: SG2: AAS: 1: 1: demo11232322<br>http://www.zvei.de/SG2/aas/1/1/demo11232322                       |
| Submodelo de identificación (tipo) | Identificación del tipo de submodelo            | submodelos son seleccionados base, otros gratuita | urn: GMA: 7,20: contractnegotiation: 1: 1<br>http://www.vdi.de/gma720/contractnegotiation/1/1                |
| Submodelo ID (Instancia)           | La identificación de la instancia del submodelo | Gratis  | urn: GMA: 7,20: contractnegotiation: 1: 1 # 001<br>http://www.vdi.de/gma720/ contractnegotiation / 1/1 # 001 |

<sup>3</sup> URL son también URI

|   |  |                        |   |
|---|--|------------------------|---|
| ID de Propiedad / Parámetro / tipo de estado  | La identificación de los bienes, los tipos de parámetros y de estado         | Específico del dominio | urn: PROFIBUS: PROFIBUS-PA: V302: Parámetro: 1: 1: MaxTemp<br>http://www.zvei.de/SG2/aas/1/1/demo11232322/ma xtemp                    |
| ID de Propiedad / Parámetro / estado de instancia<br><br>(No se utiliza por metamodelo) | La identificación de la instancia de la propiedad, y el estado del parámetro | Específico del dominio | urn: PROFIBUS: PROFIBUS-PA: V3-02: Parámetro: 1: 1: MaxTemp # 0002<br><br>http://www.zvei.de/SG2/aas/1/1/demo11232322/ma xtemp # 0002 |

**Nota:** la última fila de la tabla sólo se utiliza para la terminación; metamodelo no prevé identificadores para p instancias ROPIEDAD / parámetro / de estado.

### 3.3.7 Creación de una instancia submodelo basado en un tipo de sub-modelo existente

Con el fin de crear una instancia de un tipo de submodelo existente, debe haber una especificación pública del tipo submodelo, por ejemplo, mediante la publicación por Plattform Industrie 4.0. Como caso especial, crear instancias de un submodelo de un tipo submodelo no pública, tales como las especificaciones del fabricante, también es posible.

En noviembre de 2018, no hay tipos de submodelos estandarizados finalmente publicados disponibles, pero algunos ejemplos se describen en [6], que proporciona tablas simples con viviendas en una jerarquía predefinida. En cada tipo de submodelo, la Identificadores de definiciones de propiedades para ser utilizados como referencias semánticas ya están predefinidos. Una instanciación de tal submodelo sólo tiene que crear propiedades con una referencia semántica a las definiciones de propiedades y adjuntar los valores de estas propiedades.

En tal caso, también se predefine el Identificador del tipo de submodelo existente, probablemente como una dirección URL, y es para ser utilizado como referencia semántica para la instancia submodelo.

Lo que queda es crear un identificador de la propia instancia submodelo, que es en el caso regular y URI / URL.

**Nota:** para mantener la integridad a través de múltiples Shells Administración, referenciación apropiado ( derivado de) entre instancias submodelos y tipos submodelo tiene que ocurrir, así como para las instancias de submodelos de tipos y casos activos Interl entintado. Un marco posible podría entonces controlar y sincronizar los cambios a los estados de valor de las instancias de submodelos de acuerdo con los requisitos del usuario (no siempre se desea la sincronización automática).

### 3.3.8 Se pueden formar nuevos submodelos o de propiedad?

Es en el interés de Industrie 4.0 para tantas submodelos como sea posible, incluyendo submodelos libres y propietarios, que se formen ( • [ 4], "conjuntos de propiedades libre").

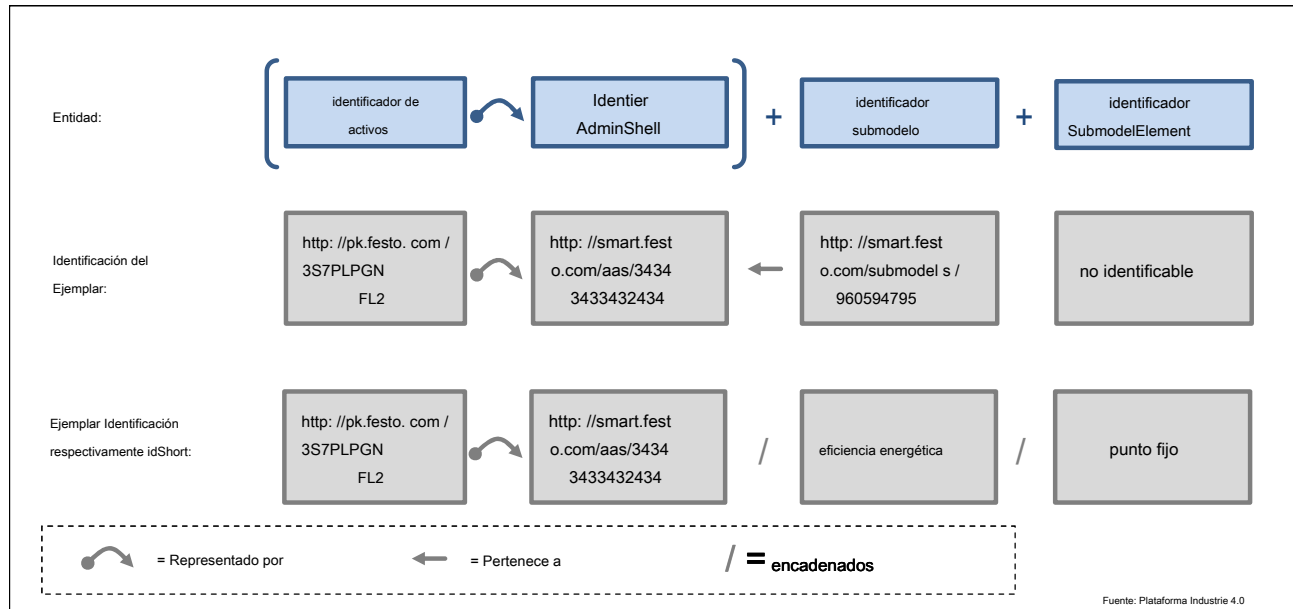
Un submodelo se puede formar en cualquier momento por un Shell de administración específica de un activo. Para este fin, el proveedor de la Concha de administración puede formar identificadores internos para el tipo y la instancia del submodelo de acuerdo con la Sección 3.3.5. Todos los sistemas I4.0 están llamados a ignorar submodelos y propiedades que no se conocen de forma individual, y simplemente a "pasar por alto" de ellos. Por esta razón, siempre es posible depositar propiedad - por ejemplo específico del fabricante o específica del usuario - la información, submodelos o propiedades en una cáscara de Administración.

**Nota:** es en la intención de la Administración de Shell, que la información patentada está incluido también. Por ejemplo, para enlace a esquemas de identificación a nivel de empresa o información requeridos para el procesamiento de datos en toda la empresa. Por esto, una única infraestructura puede ser utilizado para transportar información estandarizada y de propiedad exclusiva, al mismo tiempo; esto transmite la introducción (y más tarde la normalización) de los nuevos elementos de información también. Nota: si una instancia submodelo se forma sin una clara relación con un tipo submodelo o la definición semántica, esto será de uso limitado para otros usuarios / acceso a los sistemas de la Concha de Administración, ya que estos no pueden captar el contexto semántico de los datos contenidos.

## 3.3.9 El uso de ID corta para las entidades identificables

El Shell de administración fomenta el uso de identificadores únicos en todo el mundo en un alto grado. Sin embargo, en algunos casos, esto puede dar lugar a ineficiencias. Un ejemplo podría estar refiriéndose a una propiedad, que es parte de un submodelo que es parte de una cáscara de Administración y cada uno de ellos identificado por identificadores globales [4]. Por ejemplo, en una aplicación que ofrece una arquitectura orientada a los recursos (ROA), un localizador de recurso único en todo el mundo (URL) podría estar compuesto por una serie de segmentos, que a su vez no tiene por qué ser única en el mundo:

Figura 6 La motivación de identificadores ejemplares y idShort



Con el fin de permitir que tan eficiente direccionamiento de entidades por una API de una cáscara de Administración, idShort se proporciona un conjunto de clases del metamodelo, que heredan de la clase abstracta referenciable, con el fin de hacer referencia a las entidades que estén a cargo (•

3.4). Sin embargo, se requiere un sistema externo relativo a los recursos de una cáscara de Administración para comprobar los respectivos semántica afirmando semanticId En primer lugar, antes de acceder a las entidades de carne de identidad o idShort (• 3.5.2).



Figura 7 Descripción general Metamodel de la Shell Administración de Activos

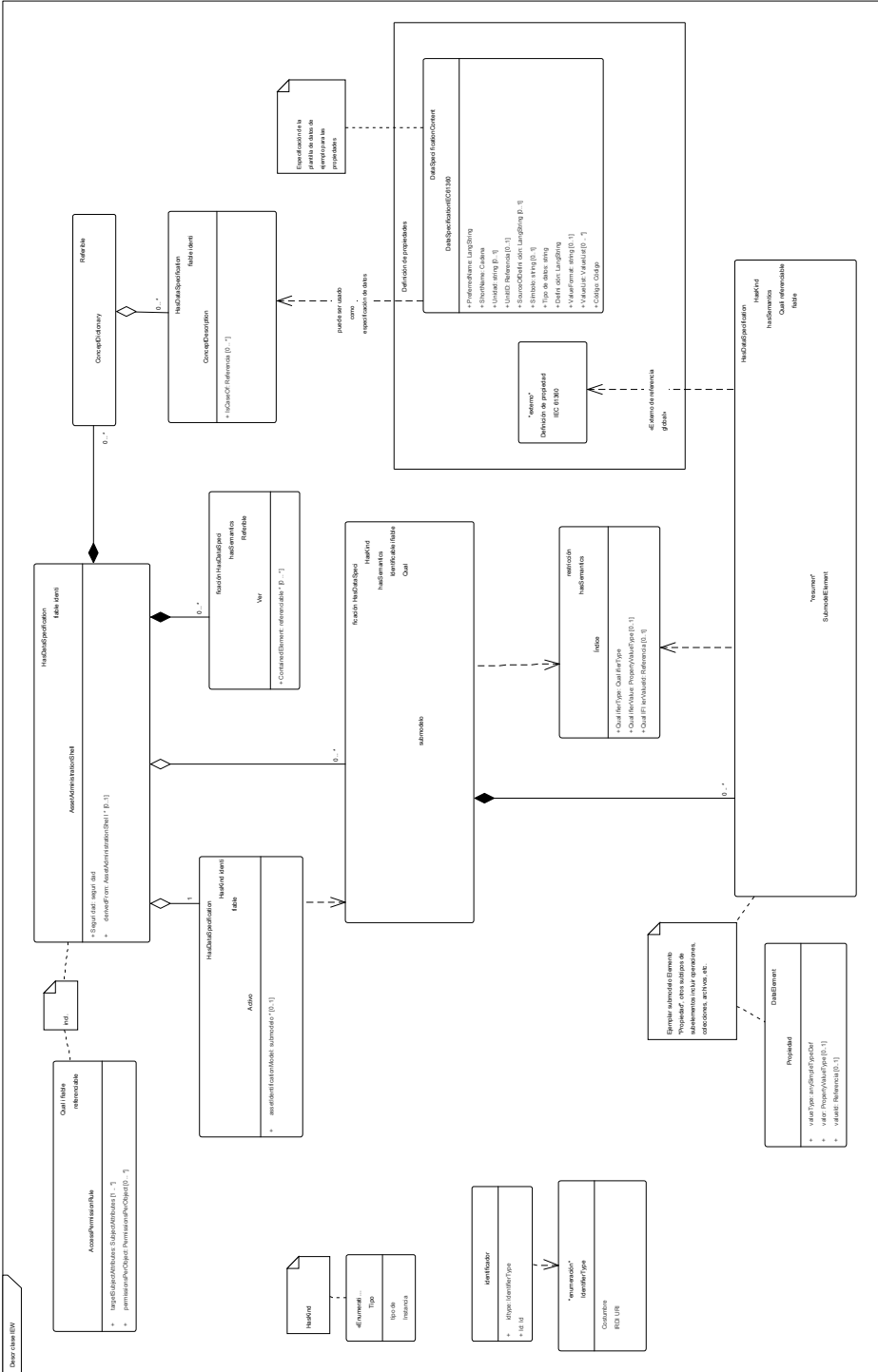
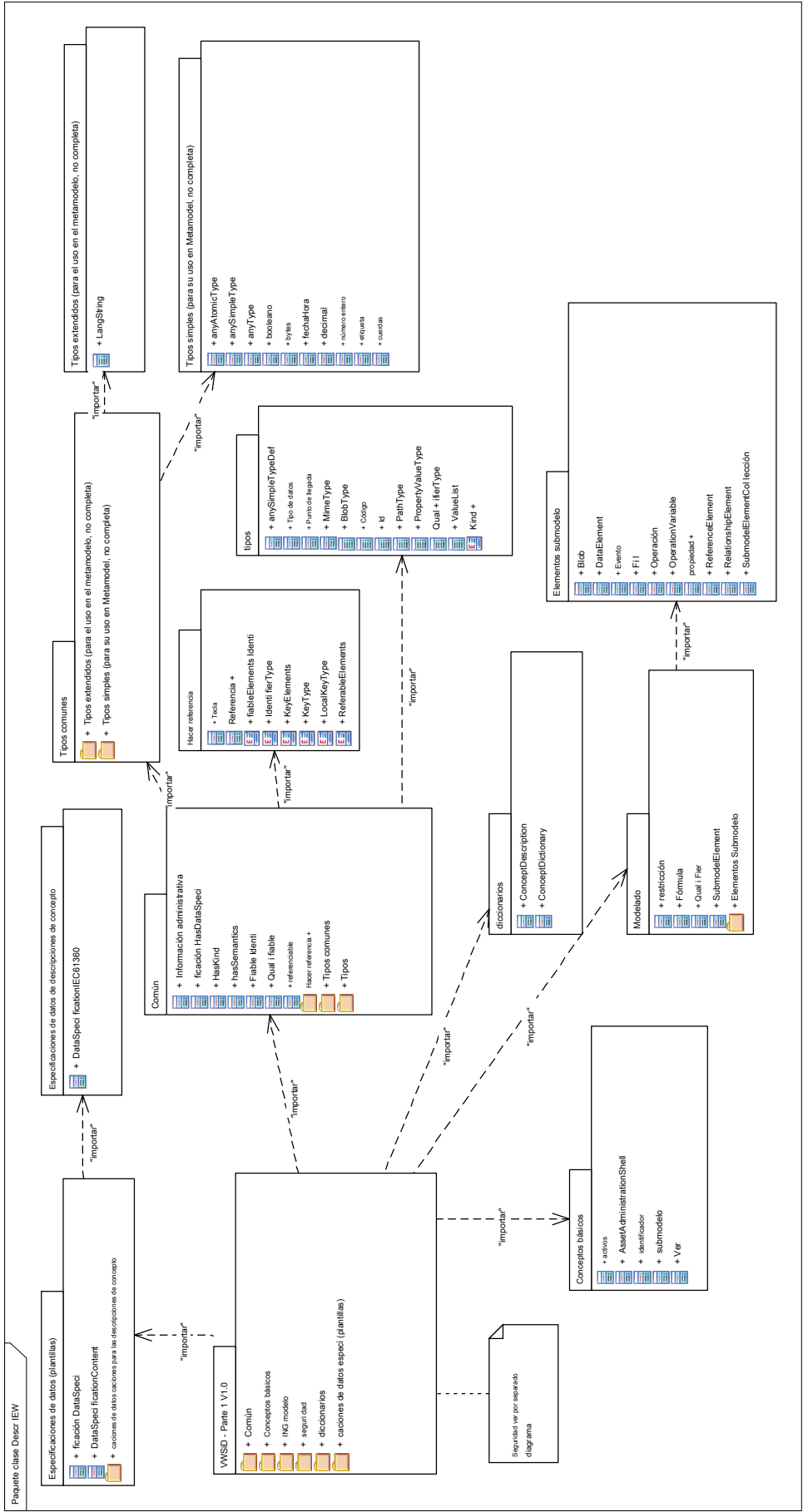


Figura 8 vista general paquete Metamodel



### 3.4 Descripción general del metamodelo de la Concha de Administración

En esta cláusula se presenta una visión general de los principales conceptos del metamodelo AssetAdministration Shell. Las partes principales de un Shell Administración de activo (AAS) es el activo que está representando así como los submodelos. Opcionalmente, los diccionarios y las vistas pueden ser parte de la AAS. Un diccionario contiene los llamados descripciones de conceptos. Para más detalles véase el numeral 3.5.3. Vistas definen un conjunto de elementos seleccionados para un grupo de interés específico. Para más detalles véase el apartado 3.5.11. Un AAS representa exactamente un activo. tipos de activos y las instancias de activos se distinguen por establecer el atributo " tipo ". Para más detalles véase el numeral 3.5.2.3.

**Nota:** el modelado UML utiliza las llamadas clases abstractas para denotar conceptos reutilizados como "hasSemantics", "Calificable", etc.

En caso de una AAS de un activo ejemplo, una referencia a la AAS que representa el tipo activo correspondiente o otra instancia activo se fue derivado de se puede añadir ( derivado de). Lo mismo vale para AAS de un tipo de activo: también tipos se pueden derivar de otros tipos.

Un activo típicamente puede ser representado por varios diferentes propiedades de identificación como por ejemplo el número de serie, su código de RFID etc. Tales propiedades de identificación locales se definen en el submodelo identificación activo ( assetIdentificationModel). Para más detalles véase el numeral 3.5.4.

EAA, activos, submodelos y descripciones de concepto tiene que ser globalmente único de identificación ( Identificable). Otros elementos como por ejemplo propiedades, diccionarios locales individuales sólo necesitan ser referible dentro del modelo y por lo tanto sólo necesitan un identificador local ( idShort desde Referible). Para más detalles sobre la identificación véase el capítulo 3.3 Identificación de las entidades. Para más detalles sobre identificable y referenciable ver 3.5.2.1.

submodelo s consisten de un conjunto de elementos de submodelos. submodelo elementos pueden ser calificados por una llamada Índice. Para más detalles véase el numeral 3.5.5.

Existen diferentes subtipos de elementos submodelos como propiedades, operaciones, colecciones etc. Para más detalles véase la cláusula 3.5.5. Un elemento submodelo típico se muestra en la figura visión general: una propiedad. Una propiedad es un elemento sub-modelo de datos que tiene un valor de tipo simple como cadena, fecha, etc Para obtener más información sobre propiedades véase la cláusula 3.5.7. Cada elemento submodelo necesita una definición semántica ( semanticId en HasSemantics). El elemento submodelo podría referirse ya sea directamente a una definición semántica correspondiente proporcionada por una referencia externa (por ejemplo, a un ss eCI @ o IEC CDD definición de propiedad) o puede hacer referencia a un elemento submodelo de tipo = Tipo que define la semántica de los elementos de submodelos de tipo = Ejemplo. Para más detalles véase el numeral 3.5.2.5.

El AAS en sí también puede definir su propio diccionario que contiene las definiciones semánticas de sus elementos de submodelos. Estas definiciones semánticas son llamados descripciones de conceptos ( ConceptDescription). Es opcional si un AAS define su propio diccionario de conceptos ( ConceptDictionary) o no. Para más detalles véase el apartado 3.5.12.

El diccionario de conceptos puede contener copias de las definiciones de propiedad de los estándares externos. En este caso, se añade una definición semántica a la norma externa ( isCaseOf). isCaseOf es una definición más formal de sourceOfDefinition esto es sólo texto.

**Nota:** en este caso, la mayor parte de los atributos son redundantes debido a que estos se definen en la norma externa. Se trata de la facilidad de uso para agregar atributos de información like PreferredName, unidad etc. La consistencia wrt a las definiciones de elementos de submodelos referenciados debe garantizarse mediante utillaje correspondiente.

El diccionario concepto también puede contener definiciones de propiedad. En este caso el proveedor del AAS deberá tener en cuenta que hay interoperabilidad con otros miembros de AA se puede asegurar. Plantillas de especificación de datos ( hasDataSpecification) se puede utilizar para definir los atributos que (además de los predefinida por el metamodelo) se utiliza para definir un elemento submodelo o una descripción concepto. Para la descripción de las propiedades concepto general de la plantilla de especificación de datos siguiendo la norma IEC 61360 se utiliza. Para que denota plantillas de especificación de datos que se recomienda utilizar el << >> plantilla - se utiliza la dependencia. Para más detalles véase el numeral 3.5.2.6.

Algunas plantillas de especificación de datos como la plantilla para IEC 61360 (definiciones de propiedades DataSpecification\_IEC61360) están predefinidos de forma explícita y recomendado para ser utilizado por la Plataforma Industrie 4.0. Para más detalles véase el numeral 3.6.2. Si se utilizan plantillas de propiedad, de nuevo, la interoperabilidad con otras AAS no puede garantizarse. Además de elementos de submodelos incluyendo propiedades y descripciones concepto también otros elementos identificables pueden utilizar plantillas adicionales ( HasDataSpecification). Para más detalles véase el numeral 3.5.2.7. elementos de submodelos y los submodelos sí mismos pueden tener calificadores adicionales ( Calificable). Por calificable puede haber más de un calificador. Para más detalles véase el numeral 3.5.2.6.

Además, Las vistas pueden ser definidas dentro de un AAS. Vistas puede consistir en cualquier elementos que son referibles ( containedElement). Una "Visión Seguridad", por ejemplo, contiene todas las propiedades o las operaciones que son relevantes para la seguridad y que necesitan un tratamiento especial. Para más detalles véase el apartado 3.5.11. Una definición de la vista también se puede utilizar en diferentes etapas del ciclo vital. Por ejemplo, podría haber un punto de vista de la ingeniería y de todos los objetos referenciados se eliminan antes de entregar el AAS al cliente.

Por cada AAS aspectos de seguridad deben tenerse en cuenta ( seguridad). En este documento, el aspecto de control de acceso se explica con más detalle. Las denominadas reglas de permiso de acceso se definen, que definen qué permiso autenticado un tema específico sobre el cual tiene objeto. Para más detalles véase el capítulo cláusula.

La Figura 8 da una imagen completa de todos los elementos definidos en el metamodelo excluida la seguridad. La seguridad se encuentra en la cláusula 5.3.

### 3.5 Metamodelo detalles Especificación: Los designadores

#### 3.5.1 Introducción

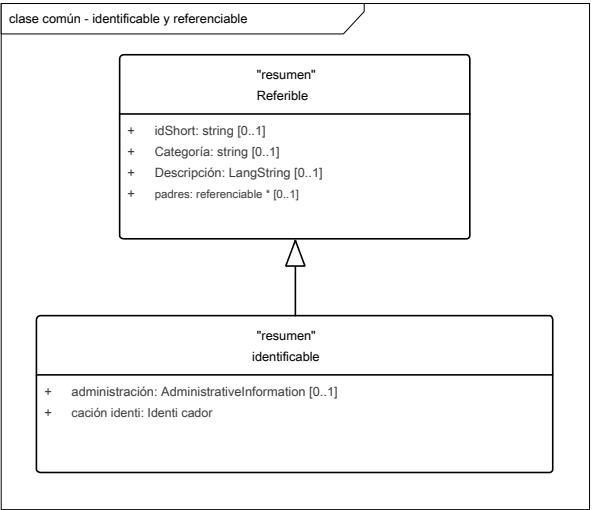
En esta cláusula las clases del metamodelo se especifican en detalle. En el Anexo B de la plantilla que se utiliza para describir las clases y las relaciones se explica. En el anexo D algunos de los diagramas se muestran junto con todos sus atributos heredados para dar una visión completa.

Para la comprensión de las especificaciones, es crucial para entender los atributos comunes en primer lugar (cláusula 3.5.2). Ellos se vuelven a utilizar a lo largo de las especificaciones de las otras clases ( "hereda de") y definen conceptos importantes como identificable, calificable etc. Son abstracto, es decir, no hay ninguna instancia de objeto de tales clases.

#### 3.5.2 atributos comunes

##### 3.5.2.1 Identificables y Referables

Figura 9 Metamodel para identificables y Referables



El metamodelo distingue entre los elementos que son identificables, referible o ninguno de ambos. Un elemento de identificación como un identificador único global ( Identifier). elementos atribuibles pueden ser referenciados pero para hacerlo el contexto del elemento que se necesita. A referable tiene un identificador único corto ( idShort) que es única solo en su contexto, su espacio de nombres. Un identificable también es atribuible pero hay elementos que no son atribuibles: no son más que los atributos de un atribuibles. Identificables pueden tener información administrativa como la versión, etc.

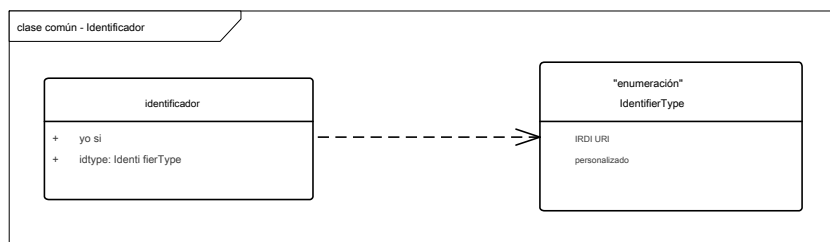
Un espacio de nombres se define como sigue en este contexto: El elemento padre un elemento es parte de y que es ya sea referible o identificable es el espacio de nombre del elemento. Ejemplos: Un submodelo es el espacio de nombres para las propiedades contenidas en él. El espacio de nombre de un elemento submodelo estando contenido en una colección elemento submodelo es la colección elemento submodelo. Sin embargo, para identificables el espacio de nombre no es importante, ya que identificables por definición tienen un identificador global.

| Clase:                     | Referable << >> abstracta   |            |          |          |
|----------------------------|---|------------|----------|----------|
| Explicación:               | Un elemento que es referible por su idShort. Esta identificación no es único a nivel mundial. Este ID es único dentro del espacio de nombres del elemento.  |            |          |          |
| Hereda de:                 | --  |            |          |          |
| Atributo (* = obligatorio) | Explicación   | Tipo       | Tipo     | Tarjeta. |
| idShort                    | <p>La identificación de la cadena del elemento dentro de su espacio de nombres. Restricción aasd-001: En caso de un elemento atribuibles al no ser un elemento identificable esta identificación es obligatoria y se utiliza para hacer referencia al elemento en su espacio de nombres.</p> <p><u>Restricción aasd-002: idShort será única característica letras, números, guión bajo (" _"); comenzando con una letra obligatoria. Constraint aasd-003: idShort deberá ser igualada entre mayúsculas y minúsculas.</u></p> <p>Nota: En caso de un elemento de identificación idShort es opcional pero se recomienda que definirse. Se puede utilizar como referencia única en su espacio de nombres y por lo tanto permite una mejor facilidad de uso y una aplicación más performante. En este caso, es similar a la "BrowserPath" en OPC UA.</p> <p>Nota: En caso de que el elemento es una propiedad y la propiedad tiene una <b>definición semántica ( hasSemantics) la idShort suele ser idéntico al nombre corto en Inglés.</b></p> | cuerda     | attr     | 0..1     |
| categoría                  | <p>La categoría es un valor que da más WRT meta-información a la clase de elemento. Afecta a la existencia esperado de atributos y la aplicabilidad de las limitaciones.</p> <p>Nota: La categoría no es idéntica a la definición semántica (hasSemantics) de un elemento. La categoría por ejemplo, podría denotar que el elemento es un valor de medición mientras que la definición semántica del elemento denotaría que es la temperatura medida.</p>   | cuerda     | attr     | 0..1     |
| descripción                | <p>Descripción o comentarios sobre el elemento. La descripción se puede proporcionar en varios idiomas.</p>   | LangString | attr     | 0..1     |
| padre                      | <p>Referencia al siguiente elemento padre del elemento atribuibles.</p> <p><u>Restricción Aasd-004: Añadir los padres en caso de elementos no identificables.</u></p> <p>Nota: Este elemento se utiliza para facilitar la navegación en el modelo y por lo tanto permite una aplicación más performante. En no da ninguna información adicional.</p>  | Referible  | árbitro* | 0..1     |

| Clase:                     | Identificable << >> abstracta  |                                |      |          |
|----------------------------|--|--------------------------------|------|----------|
| Explicación:               | Un elemento que tiene un identificador único global.   |                                |      |          |
| Hereda de:                 | Referible  |                                |      |          |
| Atributo (* = obligatorio) | Explicación  | Tipo                           | Tipo | Tarjeta. |
| administración             | información administrativa de un elemento de identificación.<br><br>Nota: Parte de la información administrativa como el número de versión que tenga que ser parte de la identificación. | AdministrativeInformation attr |      | 0..1     |
| identificación*            | La identificación único global del elemento.   | identificador                  | attr | 1        |

### 3.5.2.2 identificador

Figura 10 Metamodel para Identificador



Información sobre la identificación se puede encontrar en el capítulo 3.3 Identificación de las entidades. En el capítulo 3.3.4 limitaciones y recomendación sobre cuándo se debe utilizar el tipo de identificador puede ser encontrado.

Ejemplos de identificadores se pueden encontrar en el capítulo 3.3.3 Identificadores para Activos y Fundas para su administración. Véase el capítulo 3.5.2.2

Identificador de información que se apoyan tipos de identificadores.

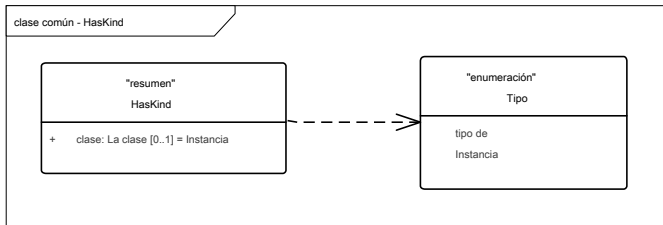
| Clase:                     | identificador  |                    |      |          |
|----------------------------|--|--------------------|------|----------|
| Explicación:               | Se utiliza para identificar de forma única una entidad mediante el uso de un identificador.  |                    |      |          |
| Hereda de:                 | --   |                    |      |          |
| Atributo (* = obligatorio) | Explicación  | Tipo               | Tipo | Tarjeta. |
| tipo de identificación*    | Tipo del identificador, por ejemplo, URI, IRDI etc. Los tipos de identificadores soportados se definen en la enumeración "IdentifierType". | IdentifierType     | attr | 1        |
| carne de identidad*        | Identificador del elemento. Su tipo se define en idtype.   | Carné de identidad | attr | 1        |

|              |  |
|--------------|--|
| Enumeración: | IdentifierType   |
| Explicación: | La enumeración de los diferentes tipos de Identificadores de identificación global |
| Literal      | Explicación  |

|               |  |
|---------------|--|
| IRDI          | IRDI según ISO29002-5 como un esquema de identificador para las propiedades y clasificaciones. |
| URI           | URI  |
| Personalizado | identificadores personalizados como GUID (identificadores únicos globales)                     |

3.5.2.3 Tiene Tipo Tipo o Instancia

Figura 11 Metamodel para HasKind



|                            |   |      |      |          |
|----------------------------|---|------|------|----------|
| Clase:                     | HasKind   |      |      |          |
| Explicación:               | Un elemento con una especie es un elemento que puede representar un tipo o una instancia. Predeterminado para un elemento es que se está representando una instancia. |      |      |          |
| Hereda de:                 | --  |      |      |          |
| Atributo (* = obligatorio) | Explicación   | Tipo | Tipo | Tarjeta. |
| Tipo                       | Kind del elemento: cualquier tipo o una instancia.<br><br>Valor por defecto = Instancia   | Tipo | attr | 0..1     |

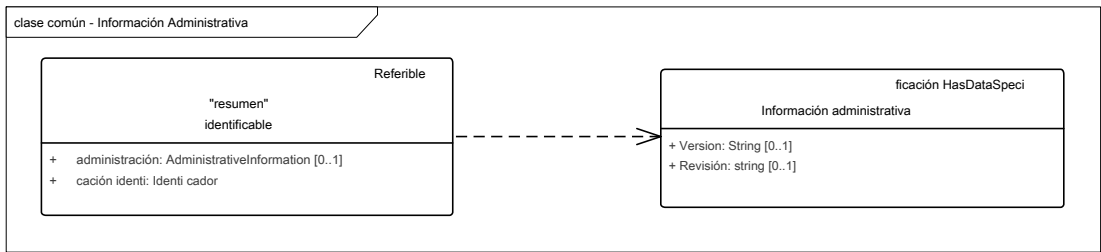
El tipo de enumeración se utiliza para indicar si un elemento es de tipo Type o Instancia.

|              |   |
|--------------|---|
| Enumeración: | Tipo  |
| Explicación: | Enumeración para que indica si un elemento es un tipo o una instancia.  |
| Hereda de:   | --  |
| Líteral      | Explicación   |
| Tipo         | hardware o software elemento que especifica los atributos comunes compartidos por todas las instancias del tipo de<br><br>[FUENTE: IEC TR 62390: 2005-01, 3.1.25]   |
| Ejemplo      | hormigón componente, claramente identificable de un cierto tipo<br><br>Nota: Se convierte en una entidad individual de un tipo, por ejemplo un dispositivo, mediante la definición de los valores de una propiedad.<br><br>Nota: En una vista orientada a objetos, un ejemplo denota un objeto de una clase (de un tipo).<br><br>[SOURCE: IEC 62890: 2016, 3.1.16] 65/617 / CDV |

Para obtener más información de los tipos e instancias ver 3.23.2 Tipos e instancias.

3.5.2.4 Información administrativa

Figura 12 Metamodel para información administrativa



Cada identificable puede tener información administrativa. Información administrativa incluye, por ejemplo

- Información sobre la versión del elemento
- Información sobre el que ha creado o que hizo el último cambio en el elemento
- Información sobre los idiomas disponibles en caso de que el elemento contiene texto, para trasladar propuso también el idioma principal o defecto puede ser definido

En la primera versión del metamodelo AAS única información de la versión según la definición de la norma IEC 61360 se define. En versiones posteriores se pueden añadir atributos adicionales. Versión corresponde en principio a la version\_identifier según la norma IEC 62832, pero no se utiliza para los identificadores de concepto solamente (IEC TS 62832-1), pero para todos los elementos identificables. Versión y revisión juntos corresponden al número de versión de acuerdo con IEC 62832.

Información administrativa permite el uso de plantillas ( HasDataSpecification) pero no hay plantillas predefinidas en esta versión del metamodelo.

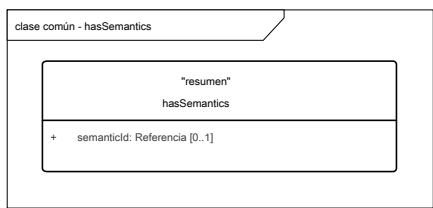
Nota: Algunos de los órganos de administración de información l igual que el número de versión puede ser que necesite para ser parte de la identificación.

|                            |  |        |      |          |
|----------------------------|--|--------|------|----------|
| Clase:                     | Información administrativa   |        |      |          |
| Explicación:               | metainformación administrativo para un elemento como información de versión.   |        |      |          |
| Hereda de:                 | HasDataSpecification   |        |      |          |
| Atributo (* = obligatorio) | Explicación  | Tipo   | Tipo | Tarjeta. |
| versión                    | Versión del elemento.  | cuerda | attr | 0..1     |
| revisión                   | Revisión del elemento.<br><br>Restricción aasd-005: Una revisión requiere una versión.<br>Esto significa que, si no hay una versión que no hay ninguna revisión. | cuerda | attr | 0..1     |



3.5.2.5 Las referencias semánticas

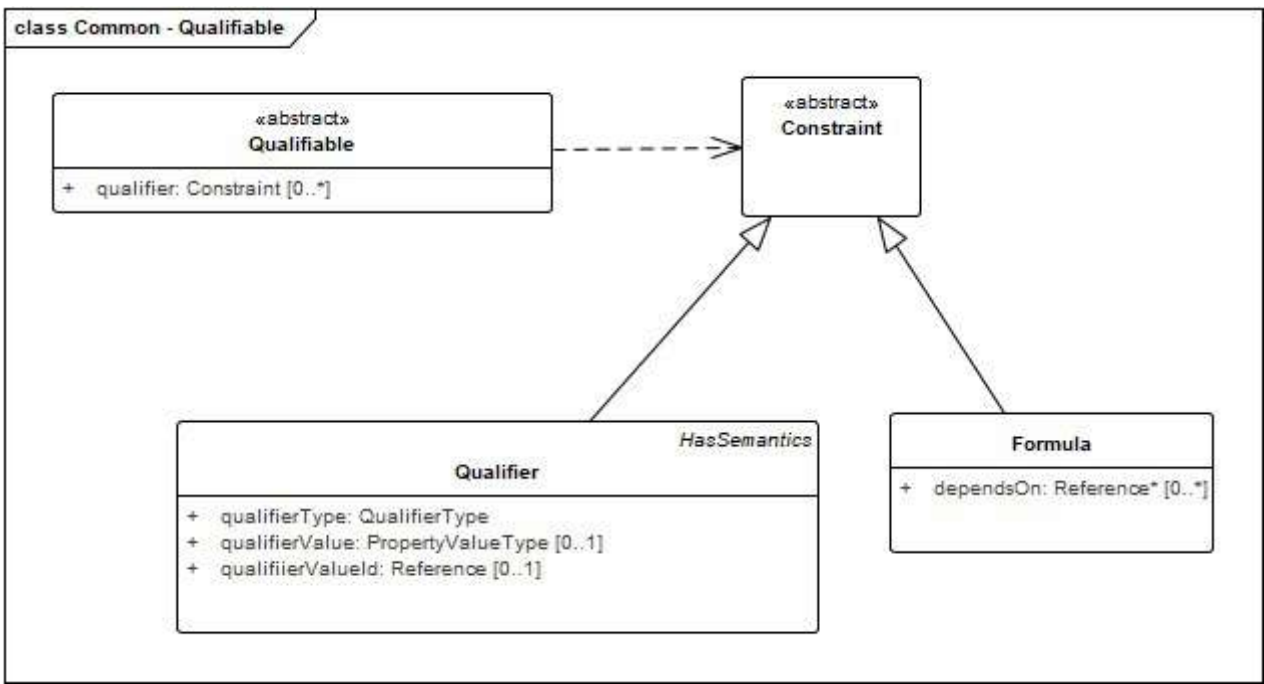
Figura 13 Metamodel para Referencias semánticas (hasSemantics)



|                            |   |            |      |          |
|----------------------------|---|------------|------|----------|
| Clase:                     | HasSemantics << >> abstracta  |            |      |          |
| Explicación:               | Elemento que puede tener una definición semántica.  |            |      |          |
| Hereda de:                 | --  |            |      |          |
| Atributo (* = obligatorio) | Explicación   | Tipo       | Tipo | Tarjeta. |
| semanticId                 | Identificador de la definición semántica del elemento. Se llama Identificación del semántica del elemento.<br><br>La identificación semántica o bien puede hacer referencia a una identificación global externa o puede hacer referencia a un elemento del modelo de tipo atribuibles = Tipo que define la semántica del elemento.<br><br>Nota: En muchos casos, el idShort es idéntico al nombre corto dentro de la definición semántica como se indica a través de esta identificación semántica. | Referencia | attr | 0..1     |

3.5.2.6 Qualifiables

Figura 14 Metamodel Qualifiables y limitaciones



Para los elementos calificables calificadores adicionales pueden ser definidos. Para más detalles sobre los calificadores y para los tipos predefinidos de clasificación estandarizados véase la Norma IEC 62569-1. Por ejemplo, un calificador de nivel que define el tipo de nivel valor mínimo, valor máximo, valor típico y el valor nominal se puede encontrar en la norma IEC 62569-1. se planifican tipos clasificatorios adicionales que se define en el trabajo en curso de la norma DIN 92000 SPEC como por ejemplo expresiones semántica y lógica expresión. Si no hay tipos de clasificación predefinidas o la cualificación adicional es bastante complejo entonces en lugar de un conjunto de calificadores también una fórmula puede ser definido.

En la figura 15 un ejemplo de una fórmula dependiendo de la propiedad "Estado" se muestra. Hasta ahora ningún lenguaje de fórmulas se define para el AAS.

Figura 15 Ejemplo Fórmula

class Common - Fórmula Example

Example for Formula (**Status != RUNNING**):

```
<aas:Formula>
  <aas:dependsOn>
    <Keys> <Key local="true" type="AssetAdministrationShell" keyType="URI">http://myshell</Key>
          <Key local="true" type="Submodel" keyType="Local">Machine</Key>
          <Key local="true" type="Property" keyType="Local">Status</Key> </Keys>
    <aas:dependsOn>
      != RUNNING
  </aas:Formula>
```

|                            |   |             |      |          |
|----------------------------|---|-------------|------|----------|
| Clase:                     | Calificable << >> abstracta   |             |      |          |
| Explicación:               | El valor de un elemento calificable puede ser calificado adicionalmente por uno o más calificadores o fórmulas complejas. |             |      |          |
| Hereda de:                 | --  |             |      |          |
| Atributo (* = obligatorio) | Explicación   | Tipo        | Tipo | Tarjeta. |
| Índice                     | calificación adicional de un elemento de -qualifiable.  | Restricción | aggr | 0 .. *   |

|                            |   |      |      |          |
|----------------------------|---|------|------|----------|
| Clase:                     | Restricción << >> abstracta                                   |      |      |          |
| Explicación:               | Una restricción se utiliza para calificar además un elemento. |      |      |          |
| Hereda de:                 | --  |      |      |          |
| Atributo (* = obligatorio) | Explicación   | Tipo | Tipo | Tarjeta. |

|                            |   |               |      |          |
|----------------------------|---|---------------|------|----------|
| Clase:                     | Índice  |               |      |          |
| Explicación:               | Un calificador es un par de tipo de valor que hace que las declaraciones adicionales WRT el valor del elemento. |               |      |          |
| Hereda de:                 | Restricción, hasSemantics   |               |      |          |
| Atributo (* = obligatorio) | Explicación   | Tipo          | Tipo | Tarjeta. |
| qualifierType *            | los <b>qualifierType</b> describe el tipo de la clasificación que se aplica al elemento.                        | QualifierType | attr | 1        |

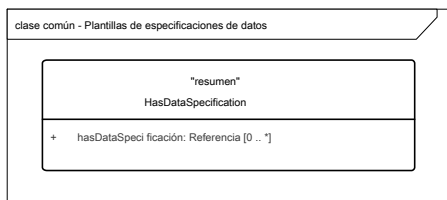
| Clase:           | Índice   |                        |     |      |
|------------------|--|------------------------|-----|------|
| qualifierValue   | El calificador es el valor del calificador.<br><br><b>Restricción Aasd-006:</b> Si ambos, el valor y la valueId están presentes, entonces el valor tiene que ser idéntica a la nombre corto del valor codificado referencia en qualifierValueId. | attr PropertyValueType |     | 0..1 |
| qualifierValueId | Referencia al ello unique global de un valor codificado.   | Referencia             | atr | 0..1 |

| Clase:                     | Fórmula  |            |      |          |
|----------------------------|--|------------|------|----------|
| Explicación:               | Una fórmula se utiliza para describir las limitaciones por una expresión lógica.   |            |      |          |
| Hereda de:                 | Restricción  |            |      |          |
| Atributo (* = obligatorio) | Explicación  | Tipo       | Tipo | Tarjeta. |
| depende de                 | Una fórmula puede depender de elementos referibles o incluso externos globales - asumidos que pueden ser referenciados y su valor puede ser evaluado - que se utilizan en la expresión lógica. | Referencia | aggr | 0 .. *   |

--

### 3.5.2.7 Plantilla para la especificación de datos

Figura 16 Metamodel para HasDataSpecification



| Clase:                     | HasDataSpecification << >> abstracta   |            |      |          |
|----------------------------|--|------------|------|----------|
| Explicación:               | Elemento que puede tener plantillas de especificación de datos. Una plantilla define los atributos adicionales un elemento puede o debe tener. |            |      |          |
| Hereda de:                 | --   |            |      |          |
| Atributo (* = obligatorio) | Explicación  | Tipo       | Tipo | Tarjeta. |
| hasDataSpecification       | referencia Global a la plantilla de especificación de datos utilizado por el elemento.   | Referencia | aggr | 0 .. *   |

```

classDiagram
    class AssetAdministrationShell {
        + security: Security
        + derivedFrom: AssetAdministrationShell* [0..1]
    }
    class Security {
        + history: History
        + accessControlPolicyPoints: AccessControlPolicyPoints
        + trustAnchor: Certificate [0..*]
        + resetPolicy: Submodel* [0..1]
    }
    class Asset {
        + assetIdentificationModel: Submodel* [0..1]
    }
    class Submodel {
        + 
    }
    class View {
        + containedElement: Referable* [0..*]
    }
    class ConceptDictionary {
        + 
    }

    AssetAdministrationShell "1" o-- "1" Security
    AssetAdministrationShell "1" o-- "0..*" Asset
    AssetAdministrationShell "1" o-- "0..*" View
    AssetAdministrationShell "1" o-- "0..*" Submodel
    AssetAdministrationShell "1" o-- "0..*" ConceptDictionary
    AssetAdministrationShell ..> Security
    AssetAdministrationShell ..> Asset
    AssetAdministrationShell ..> Submodel
    Security ..> AssetAdministrationShell
    Security ..> Submodel
    
```

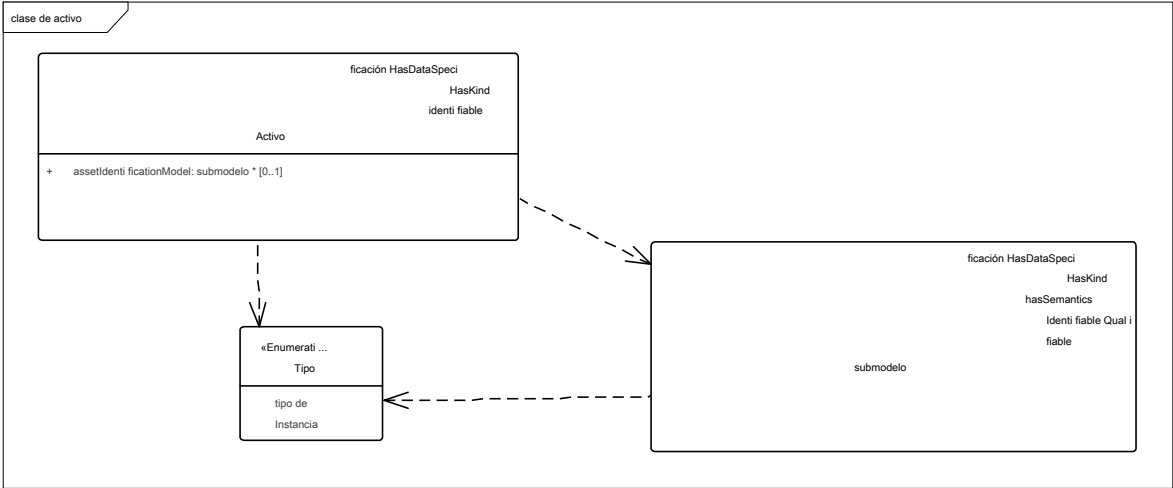
los derivado de atributo se utiliza para establecer una relación entre dos cáscaras AssetAdministration que se derivan de uno al otro. Para obtener información más detallada sobre la derivado de concepto véase la cláusula 3.2 Tipos e instancias.

| Clase:                     | AssetAdministrationShell  |                          |          |          |
|----------------------------|---|--------------------------|----------|----------|
| Explicación:               | Un Shell AssetAdministration.   |                          |          |          |
| Hereda de:                 | HasDataSpecification; identifiable;   |                          |          |          |
| Atributo (* = obligatorio) | Explicación   | Tipo                     | Tipo     | Tarjeta. |
| derivado de                | La referencia a la AAS el AAS se deriva de.   | AssetAdministrationShell | árbitro* | 0..1     |
| seguridad*                 | Definición de los aspectos relevantes de seguridad de la AAS.   | Seguridad                | aggr     | 1        |
| activo*                    | El activo el AAS está representando.  | Activo                   | árbitro* | 1        |
| submodelo                  | El activo de un AAS se describe típicamente por uno o más submodelos.<br><br>Temporalmente no submodelo podría ser asignado a la AAS.   | submodelo                | árbitro* | 0 .. *   |
| conceptDictionary          | Un AAS max tiene uno o más diccionarios de concepto que se le asignan. Los diccionarios concepto típicamente contienen sólo las descripciones de los elementos que también se usan en la AAS (a través HasSemantics). | ConceptDictionary        | aggr     | 0 .. *   |

|        |   |     |      |        |
|--------|---|-----|------|--------|
| Clase: | AssetAdministrationShell  |     |      |        |
| ver    | Si las vistas de interés específicos necesarios se pueden definir en los elementos de la AAS. | Ver | aggr | 0 .. * |

3.5.4 Atributos de activos

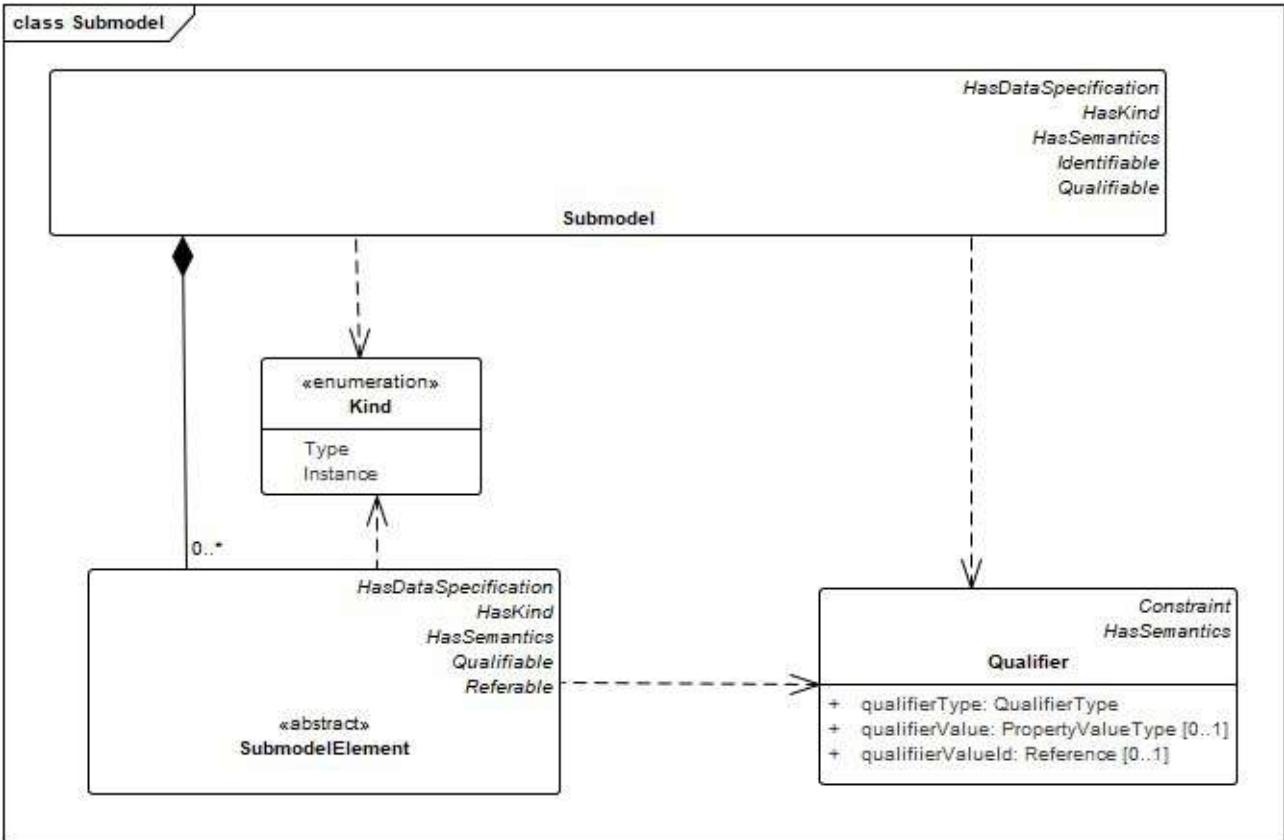
Figura 18 Metamodel de Activos



|                            |   |           |        |          |
|----------------------------|---|-----------|--------|----------|
| Clase:                     | Activo  |           |        |          |
| Explicación:               | <p>Un Activo describe los metadatos de un activo que está representado por un AAS. El activo puede representar un tipo de activo o una instancia de activos.</p> <p>El activo tiene un identificador único global plus - si es necesario - específicos de dominio adicional (patentados) identificadores.</p> |           |        |          |
| Hereda de:                 | HasDataSpecification; identificable; HasKind  |           |        |          |
| Atributo (* = obligatorio) | Explicación   | Tipo      | Tipo   | Tarjeta. |
| assetIdentificationModel   | Una referencia a una submodelo que define el manejo de los identificadores adicionales específicos de dominio (patentados) para el activo como por ejemplo, número de serie, etc.   | submodelo | árbol* | 0..1     |

3.5.5 Submodelo y submodelo atributos de elemento

Figura 19 Metamodel para Submodelo



|                            |   |                 |      |          |
|----------------------------|---|-----------------|------|----------|
| Clase:                     | submodelo   |                 |      |          |
| Explicación:               | <p>Un submodelo define un aspecto específico del activo representado por la AAS.</p> <p>Un submodelo se utiliza para estructurar la representación virtual y funcionalidad técnica de una cáscara de Administración en partes distinguibles. Cada submodelo refiere a un dominio bien definido o materia. Submodelos pueden convertirse en tipos submodelos estandarizados y por lo tanto convertirse. Submodelos pueden tener diferentes ciclos de vida.</p> |                 |      |          |
| Hereda de:                 | HasDataSpecification; hasSemantics; identifiable; calificables; HasKind   |                 |      |          |
| Atributo (* = obligatorio) | Explicación   | Tipo            | Tipo | Tarjeta. |
| submodelElement            | Un submodelo consta de cero o más elementos de submodelos.  | SubmodelElement | aggr | 0..*     |

Un ejemplo submodelo puede hacer referencia al tipo submodelo se deriva de. Formulado de una manera técnica: semanticId

de un submodelo con tipo = Instancia puede referirse a una submodelo de tipo = Tipo (tipo heredada a través de HasKind).

Un submodelo se puede calificar ( Calificable).

elemento submodelo son elementos calificables, es decir, uno o más calificador puede ser definida para cada uno de ellos. Submodelos y elementos submodelo también pueden tener plantillas de especificación de datos definidos para ellos. Una fuerza de plantilla, por ejemplo, puede definir para reflejar algunos de los atributos como Nombre Preferido y unidad de una definición de propiedad si el AAS no contiene una descripción concepto correspondiente. De lo contrario no sólo es la definición de la propiedad referenciada por

semanticId disponible para la propiedad: la búsqueda de los atributos tiene que ser realizado en línea de una manera diferente y no está disponible sin conexión.

En caso de que el submodelo es de tipo = Tipo a continuación, los elementos de submodelos dentro del submodelo se presentan los tipos de elementos de submodelos. En caso de que el submodelo es de tipo = Instancia a continuación, sus elementos submodelos representan instancias de elementos de submodelos.

|                            |   |      |      |          |
|----------------------------|---|------|------|----------|
| Clase:                     | SubmodelElement << >> abstracta   |      |      |          |
| Explicación:               | <div>Un elemento submodelo es un elemento adecuado para la descripción y la diferenciación de los activos.</div> <div>NOTA:<br/>El concepto de tipo e instancia se aplica a submodelo elementos. Las propiedades son elementos especiales de submodelos.<br/>Los tipos de propiedad están definidos en los diccionarios (como el Diccionario de Datos Común IEC o eCI @ ss), que no tienen un valor. El tipo de propiedad (tipo = Tipo) también se conoce como tipo de elemento de datos en algunas normas.<br/><br/>Las instancias de propiedad (tipo = Instancia) y típica tienen un valor. Una instancia de la propiedad también se llama par propiedad-valor en ciertas normas.</div> |      |      |          |
| Hereda de:                 | HasDataSpecification; Referible; calificables; hasSemantics; HasKind  |      |      |          |
| Atributo (* = obligatorio) | Explicación   | Tipo | Tipo | Tarjeta. |

Figura 20 Metamodel para Tipos Submodelo Element

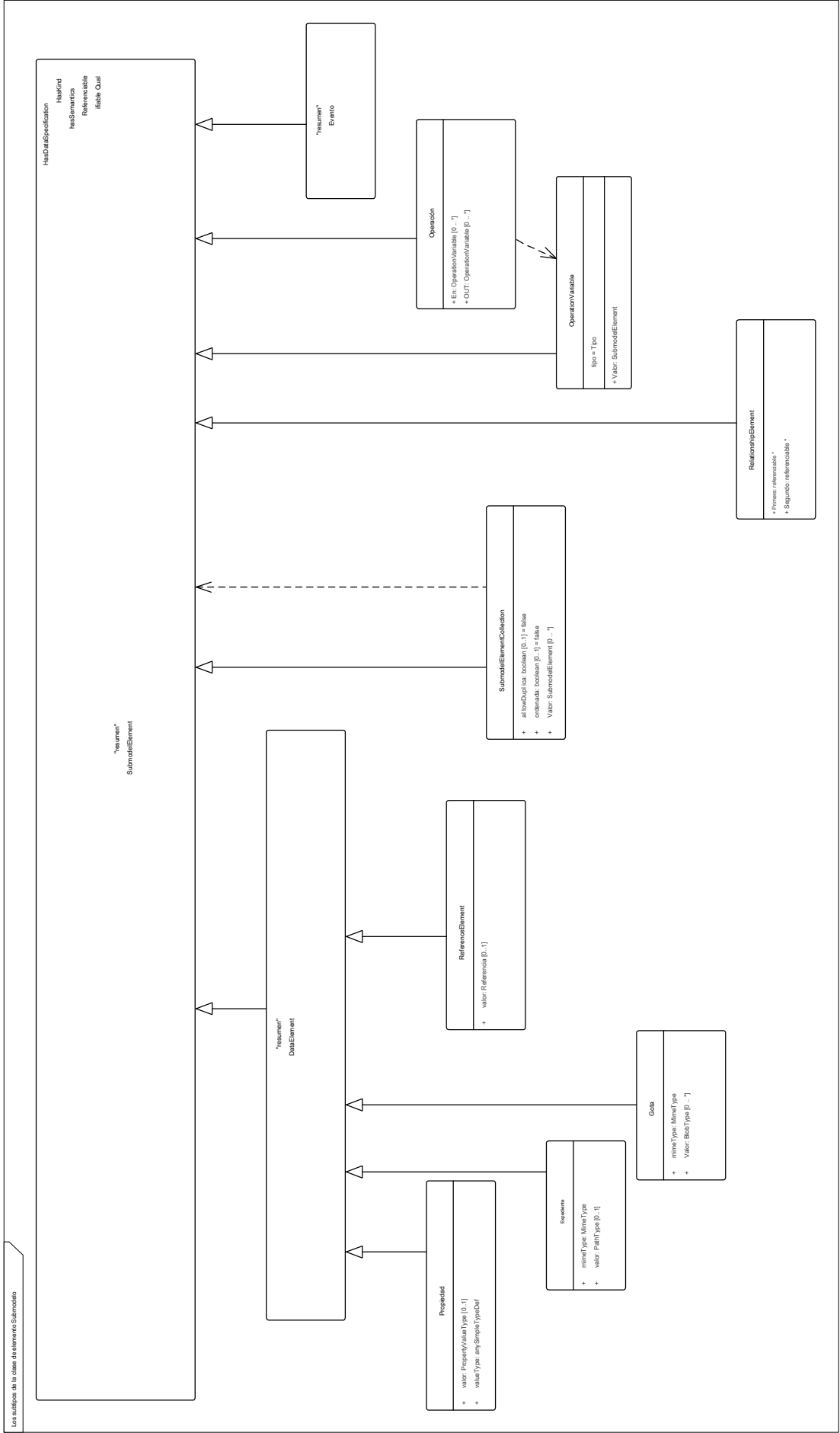
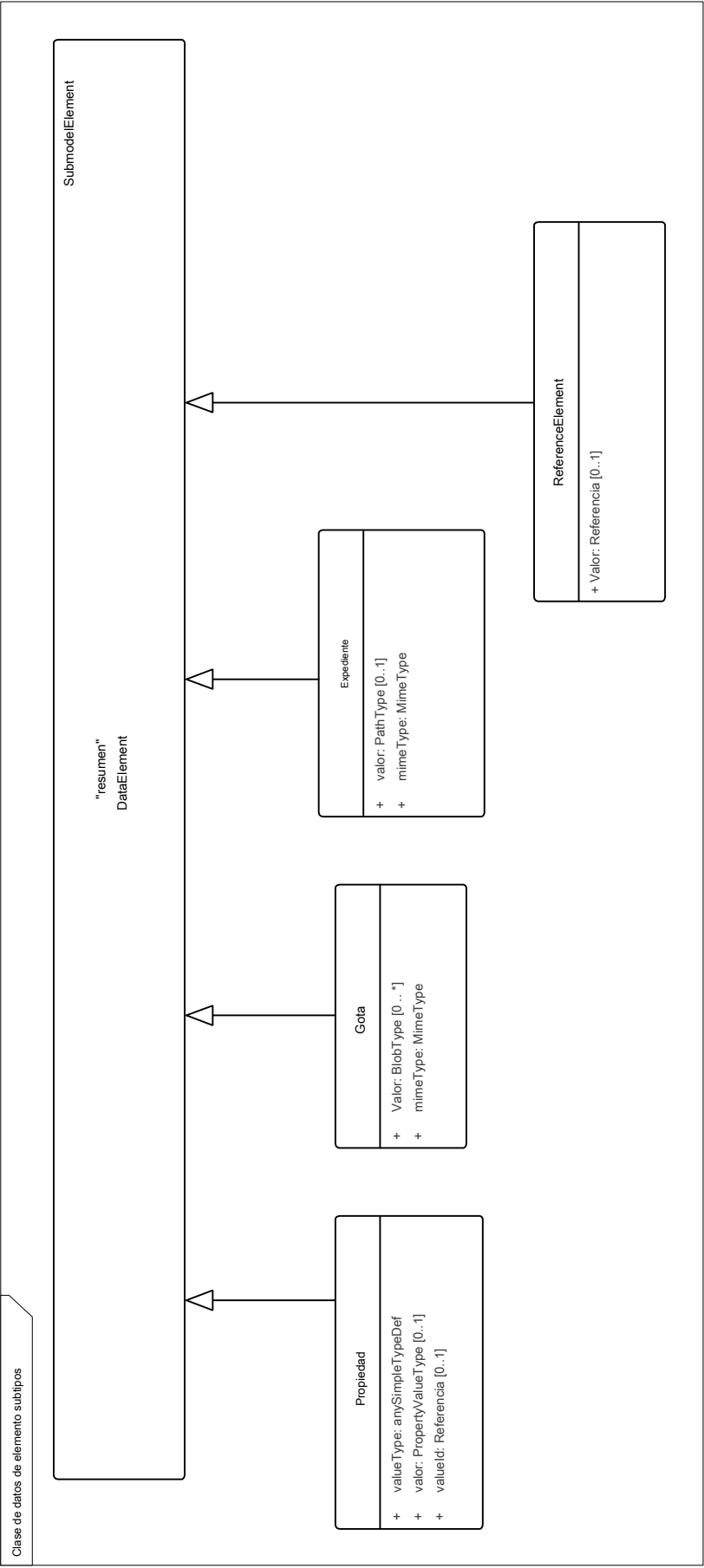




Figura 21 Metamodel para elementos de datos y sus subtipos



### 3.5.6 Resumen de los tipos de elemento submodelo

elementos submodelos incluyen propiedades de datos así como las operaciones, eventos y otros elementos necesarios para describir un modelo para un activo (véase la Figura 20).

En esta versión del metamodelo de la atención se centra en las propiedades de los datos.

### 3.5.7 Elemento de datos Atributos

Elementos de datos incluyen propiedades y los elementos de manejo de archivos y de referencia, véase la Figura 21. Las siguientes

categorías son definidas por propiedades:

| Categoría: | Aplicable a: | Explicación:   |
|------------|--------------|--|
| CONSTANTE  | Propiedad    | Una propiedad constante es una propiedad con un valor que no cambia con el tiempo.<br><br>En eCI @ ss este tipo de categoría tiene la categoría de "Codificado Valor".                 |
| PARÁMETRO  | Propiedad    | Una propiedad parámetro es una propiedad que se establece una vez y luego lo general no cambia con el tiempo.<br><br>Este es por ejemplo el caso para los parámetros de configuración. |
| VARIABLE   | Propiedad    | Una propiedad variable es una propiedad que se calcula durante el tiempo de ejecución, es decir, su valor es un valor de tiempo de ejecución.  |

| Clase:                     | DataElement << >> abstracta  |      |      |          |
|----------------------------|--|------|------|----------|
| Explicación:               | Un elemento de datos es un elemento submodelo que no se compone además de otros elementos de submodelos. Un elemento de datos es un elemento submodelo que tiene un valor. El tipo de valor es diferente para los diferentes subtipos de elementos de datos. |      |      |          |
| Hereda de:                 | SubmodelElement  |      |      |          |
| Atributo (* = obligatorio) | Explicación  | Tipo | Tipo | Tarjeta. |

| Clase:                     | Propiedad   |                   |        |          |
|----------------------------|---|-------------------|--------|----------|
| Explicación:               | Una propiedad es un elemento de datos que tiene un único valor.   |                   |        |          |
| Hereda de:                 | DataElement   |                   |        |          |
| Atributo (* = obligatorio) | Explicación   | Tipo              | Tipo   | Tarjeta. |
| valor                      | El valor de la instancia de la propiedad.<br><br><b>Restricción Aasd-007: Si ambos, el valor y la valueld están presentes, entonces el valor tiene que ser idéntica a la nombre corto del valor codificado referencia en valueld.</b> | PropertyValueType | attrqu | 0..1     |
| valueld                    | Referencia al ello unque global de un valor codificado.   | Referencia        | atr    | 0..1     |

Un tipo de medio (también de tipo MIME y el tipo de contenido) [...] es un identificador de dos partes para formatos de archivo y contenido del formato de transmisión en Internet. La Internet Assigned Numbers Authority (IANA) es la autoridad oficial para la normalización y publicación de estas clasificaciones. tipos de medios se definieron originalmente en Request for Comments 2045 en noviembre de 1996 como parte de MIME (Multipurpose Internet Mail Extensions) especificación, para denotar el tipo de contenido del mensaje de correo electrónico y archivos adjuntos; [...] por lo tanto, el tipo de nombre / MIME. <sup>4</sup>

| Clase:                     | Gola  |           |      |          |
|----------------------------|---|-----------|------|----------|
| Explicación:               | Un BLOB es un elemento de datos que representa un archivo que se encuentra con su código fuente en el atributo de valor.  |           |      |          |
| Hereda de:                 | DataElement   |           |      |          |
| Atributo (* = obligatorio) | Explicación   | Tipo      | Tipo | Tarjeta. |
| valor*                     | El valor de la instancia de BLOB de un elemento de datos blob.<br><br>Nota: En contraste con la propiedad del archivo el contenido del archivo se almacena directamente como valor en el elemento de datos Blob.  | BlobType  | attr | 0 .. *   |
| tipo MIME*                 | tipo MIME del contenido del BLOB. Los estados de tipo MIME qué archivo de la extensión del archivo tiene.<br><br>Los valores válidos son, por ejemplo "application / json", "application / xls", "image / jpg" Los valores permitidos se definen como en RFC2046. | Tipo MIME | attr | 1        |

| Clase:                     | Expediente  |           |      |          |
|----------------------------|---|-----------|------|----------|
| Explicación:               | Un archivo es un elemento de datos que representa un archivo a través de la descripción de su trayectoria.  |           |      |          |
| Hereda de:                 | DataElement   |           |      |          |
| Atributo (* = obligatorio) | Explicación   | Tipo      | Tipo | Tarjeta. |
| valor                      | Ruta y el nombre del archivo de referencia (sin extensión). La ruta puede ser absoluta o relativa.<br><br>Nota: La extensión de archivo se define mediante el uso de un calificador de tipo "MimeType". | PathType  | attr | 0..1     |
| tipo MIME*                 | tipo MIME del contenido del archivo.  | Tipo MIME | attr | 1        |

Para el manejo de archivos externos suplementarios en el intercambio de especificación de AAS en formato aasx ver también la cláusula 6.4 Convenciones para el formato de archivo del paquete de Shell Administración de Activos (AASX). Una ruta absoluta se utiliza en el caso de que el archivo existe independientemente de la AAS. Una ruta relativa, relativa a la raíz del paquete debe ser utilizado si el archivo es parte del paquete serializada del AAS.

---

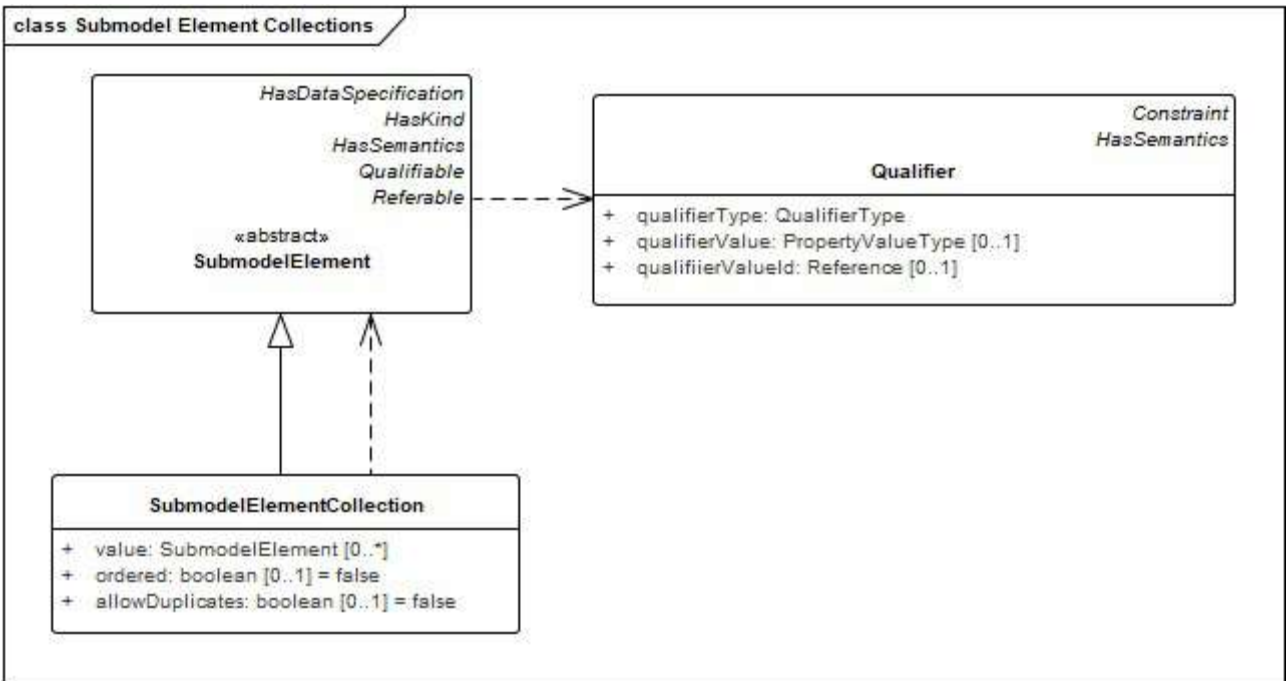
<sup>4</sup> Wikipedia.org, fecha: 09/04/2018

|                            |   |            |      |          |
|----------------------------|---|------------|------|----------|
| Clase:                     | ReferenceElement  |            |      |          |
| Explicación:               | Un elemento de referencia es un elemento de datos que define una referencia a otro elemento dentro de la misma u otra AAS o una referencia a un objeto externo o entidad. |            |      |          |
| Hereda de:                 | DataElement   |            |      |          |
| Atributo (* = obligatorio) | Explicación   | Tipo       | Tipo | Tarjeta. |
| valor                      | La referencia a cualquier otro elemento referible de la misma de cualquier otro AAS o una referencia a un objeto externo o entidad.                                       | Referencia | aggr | 0..1     |

Para más información sobre referencias véase el numeral 3.5.13.

3.5.8 Colección de elementos de datos de atributos

Figura 22 Metamodel para Submodelo Element Colecciones

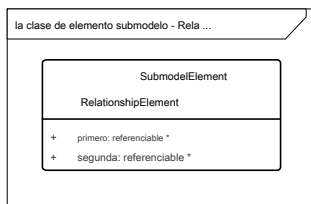


|                            |   |                 |      |          |
|----------------------------|---|-----------------|------|----------|
| Clase:                     | SubmodelElementCollection   |                 |      |          |
| Explicación:               | Una colección elemento submodelo es un conjunto o una lista de elementos de submodelos.   |                 |      |          |
| Hereda de:                 | SubmodelElement   |                 |      |          |
| Atributo (* = obligatorio) | Explicación   | Tipo            | Tipo | Tarjeta. |
| valor                      | elemento submodelo contenida en la colección.   | SubmodelElement | aggr | 0..*     |
| ordenado                   | Si se ha pedido = false entonces los elementos de la colección de propiedades no están ordenados. Si ordenado = true entonces están clasificadas los elementos de la colección. Default = false | booleano        | attr | 0..1     |

|                 |  |          |      |      |
|-----------------|--|----------|------|------|
| Clase:          | SubmodelElementCollection  |          |      |      |
|                 | Nota: Una colección de elementos ordenados submodelo se implementa normalmente en forma de matriz.                             |          |      |      |
| AllowDuplicates | Si AllowDuplicates = true entonces se permite que la colección contiene el mismo elemento varias veces.<br><br>Default = false | booleano | attr | 0..1 |

### 3.5.9 Atributos de relación

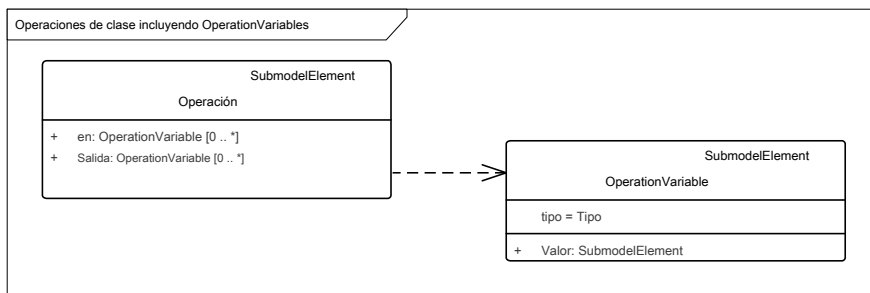
Figura 23 Metamodel de relación Elementos



| Clase:                     | RelationshipElement  |           |          |          |
|----------------------------|--|-----------|----------|----------|
| Explicación:               | Un elemento de relación se utiliza para definir una relación entre dos elementos referibles. |           |          |          |
| Hereda de:                 | SubmodelElement  |           |          |          |
| Atributo (* = obligatorio) | Explicación  | Tipo      | Tipo     | Tarjeta. |
| primero                    | El primer elemento en la relación de tomar el papel del sujeto.                              | Referible | árbitro* | 1        |
| segundo                    | Segundo elemento en la relación de tomar el papel del objeto.                                | Referible | árbitro* | 1        |

### 3.5.10 Los atributos de operación

Figura 24 Metamodel de Operaciones



| Clase:                     | Operación  |                   |      |          |
|----------------------------|--|-------------------|------|----------|
| Explicación:               | Una operación es un elemento submodelo con variables de entrada y de salida. |                   |      |          |
| Hereda de:                 | SubmodelElement  |                   |      |          |
| Atributo (* = obligatorio) | Explicación  | Tipo              | Tipo | Tarjeta. |
| en                         | parámetro de entrada de la operación.  | OperationVariable | aggr | 0..*     |

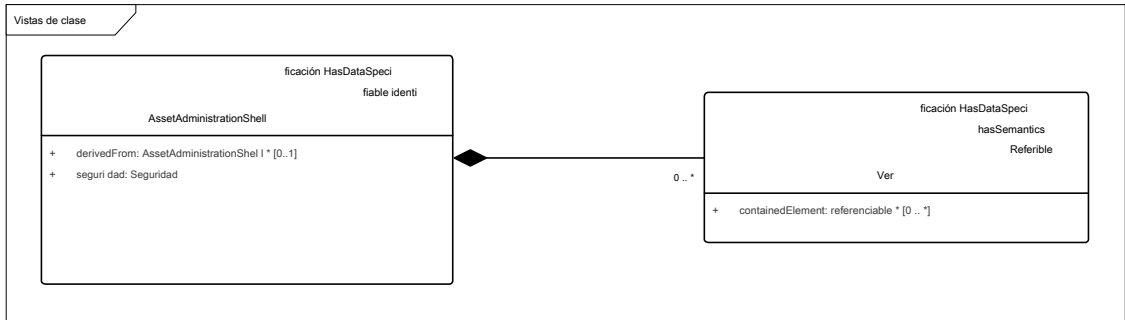
|        |                                      |                   |      |        |
|--------|--------------------------------------|-------------------|------|--------|
| Clase: | Operación                            |                   |      |        |
| afuera | Parámetro de salida de la operación. | OperationVariable | aggr | 0 .. * |

|                            |  |                 |      |          |
|----------------------------|--|-----------------|------|----------|
| Clase:                     | OperationVariable  |                 |      |          |
| Explicación:               | Una variable de funcionamiento es un elemento submodelo que se utiliza como entrada o salida variable de una operación.  |                 |      |          |
| Hereda de:                 | SubmodelElement  |                 |      |          |
| Atributo (* = obligatorio) | Explicación  | Tipo            | Tipo | Tarjeta. |
| valor*                     | Describe el argumento necesario para una operación a través de un elemento submodelo de clase = Tipo.<br><br><u>Restricción aasd-008:</u> El elemento submodelo deberá ser de tipo = Tipo. | SubmodelElement | aggr | 1        |

Nota: Las operaciones suelen especificar el comportamiento de un componente en términos de procedimientos. Por lo tanto, las operaciones permiten la especificación de servicios con las interacciones basadas en procedimientos [32].

3.5.11 Ver los atributos

Figura 25 Metamodel de Vistas



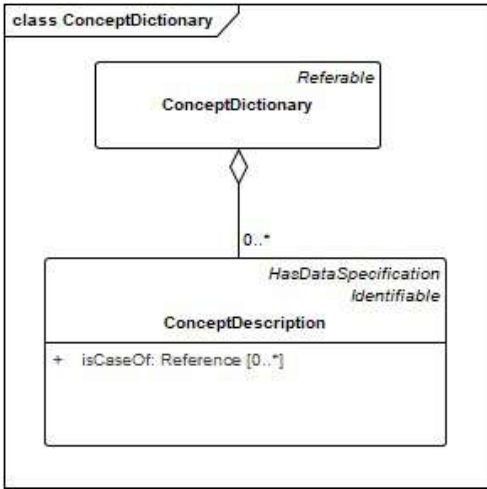
El gran número de elementos de submodelos dentro de un submodelo se puede filtrar por puntos de vista, de manera que diferentes grupos de usuarios sólo pueden ver los elementos pertinentes.

Nota: Según cláusula 1.5, las vistas son una proyección de elementos de submodelos para una perspectiva dada. Ellos no son equivalente a submodelos.

|                            |   |           |          |          |
|----------------------------|---|-----------|----------|----------|
| Clase:                     | Ver   |           |          |          |
| Explicación:               | Una vista es una colección de elementos referibles wrt a un punto de vista específico de uno o más grupos de interés. |           |          |          |
| Hereda de:                 | HasDataSpecification; Referible; hasSemantics   |           |          |          |
| Atributo (* = obligatorio) | Explicación   | Tipo      | Tipo     | Tarjeta. |
| containedElement           | elementos referibles que están contenidos en la vista.  | Referible | árbitro* | 0 .. *   |

3.5.12 Atributos ionary concepto Dict

Figura 26 Metamodel de diccionario de conceptos



|                                |   |                    |       |          |
|--------------------------------|---|--------------------|-------|----------|
| Clase:                         | ConceptDictionary   |                    |       |          |
| Explicación:                   | <p>Un diccionario contiene elementos que pueden ser reutilizados. El diccionario de conceptos contiene descripciones de conceptos.</p> <p>Normalmente, un diccionario descripción concepto de un AAS contiene sólo descripciones de concepto de los elementos utilizados en submodelos de la AAS.</p> |                    |       |          |
| Hereda de:                     | Referible   |                    |       |          |
| Atributo (* = obligatorio)     | Explicación   | Tipo               | Tipo  | Tarjeta. |
| Descripción conceptDescription | concepto define un concepto.  | ConceptDescription | árbol | 0 .. *   |

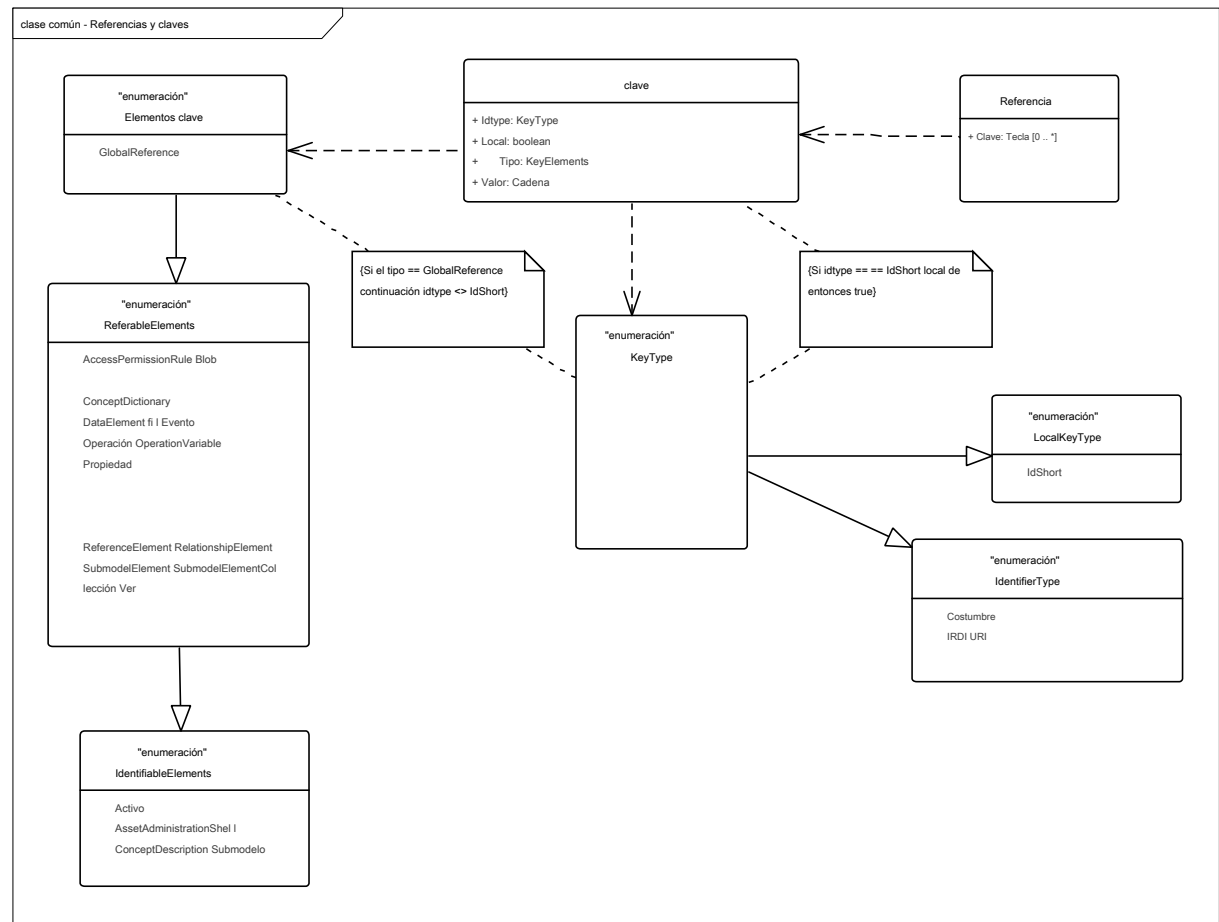
|                            |   |            |      |          |
|----------------------------|---|------------|------|----------|
| Clase:                     | ConceptDescription  |            |      |          |
| Explicación:               | <p>La semántica de una propiedad o de otros elementos que puedan tener una descripción semántica se define por una descripción concepto.</p> <p>La descripción del concepto debe seguir un esquema estandarizado (realizado como plantilla de especificación de datos).</p> |            |      |          |
| Hereda de:                 | HasDataSpecification; identificable;  |            |      |          |
| Atributo (* = obligatorio) | Explicación   | Tipo       | Tipo | Tarjeta. |
| isCaseOf                   | <p>referencia global a una definición externa del concepto es compatible o se deriva de.</p> <p>Nota: Comparar atribuir es el caso de la norma ISO 13584-32</p>   | Referencia | aggr | 0 .. *   |

Los diferentes tipos de elementos de submodelos requieren diferentes atributos para describir la semántica de ellos. Esta es la razón por una descripción concepto tiene al menos una plantilla de especificación de datos asociada a ella. Dentro de esta plantilla se definen los atributos necesarios para definir la semántica.

Véase la cláusula 3.6 para las plantillas de especificación de datos predefinidos para ser utilizada.

3.5.13 Hacer referencia en los depósitos de la Administración de Activos

Figura 27 Metamodel referencias y Keys



Nota: Las referencias se utilizan en todo el metamodelo aunque no directamente visibles.

Si un elemento no es una parte de un elemento, pero sólo hace referencia a esto se denota por un \* detrás del Tipo.

P.ej : el elemento \* significa que activo: Referencia con Key.type = Activos para la última clave en la referencia

|                            |  |       |      |          |
|----------------------------|--|-------|------|----------|
| Clase:                     | Referencia   |       |      |          |
| Explicación:               | Referencia a cualquiera de un elemento de modelo de la misma o de otra AAs o a una entidad externa. Una referencia es una lista ordenada de teclas, cada tecla hace referencia a un elemento. La lista completa de teclas puede, por ejemplo concatenar con un camino que luego le da un acceso único a un elemento o entidad. |       |      |          |
| Hereda de:                 | --   |       |      |          |
| Atributo (* = obligatorio) | Explicación  | Tipo  | Tipo | Tarjeta. |
| llave                      | de referencia único en su espacio de nombres.  | Llave | attr | 0 .. *   |

|              |  |
|--------------|--|
| Clase:       | Llave  |
| Explicación: | Una clave es una referencia a un elemento por su ID. |
| Hereda de:   | --   |



| Clase:                     | Llave  |                 |      |          |
|----------------------------|--|-----------------|------|----------|
| Atributo (* = obligatorio) | Explicación  | Tipo            | Tipo | Tarjeta. |
| tipo*                      | Denotar el tipo de entidad que se hace referencia. En <b>caso type = GlobalReference entonces el elemento es un</b> identificador único global. En todos los demás casos, las referencias clave de un elemento del modelo de la misma o de otra AAS. El nombre del elemento del modelo aparece de forma explícita. | Elementos clave | attr | 1        |
| local*                     | Indica si las referencias clave un elemento del modelo de los mismos AAS (= true) o no (= false). En el caso de locales = false La clave puede hacer referencia a un elemento del modelo de otro AAS o una entidad externa a los AA que tiene un identificador único global.                                       | booleano        | attr | 1        |
| valor*                     | El valor de la clave, por ejemplo un IRDI si el idtype = IRDI.   | cuerda          | attr | 1        |
| tipo de identificación*    | Tipo del valor llave.<br><br>En caso de idtype = idShort local debe ser verdad.<br><br>En caso tipo = GlobalReference idtype no será IdShort.  | KeyType         | attr | 1        |

La enumeración "KeyElements" es un conjunto de los siguientes valores:

- "GlobalReference"
- Todos los nombres de las clases de referables que no son identificables (ver ReferableElements enumeración en la Figura 27)
- Todos los nombres de las clases de identificables (ver IdentifiableElements enumeración en la Figura 27)

### 3.5.14 tipos

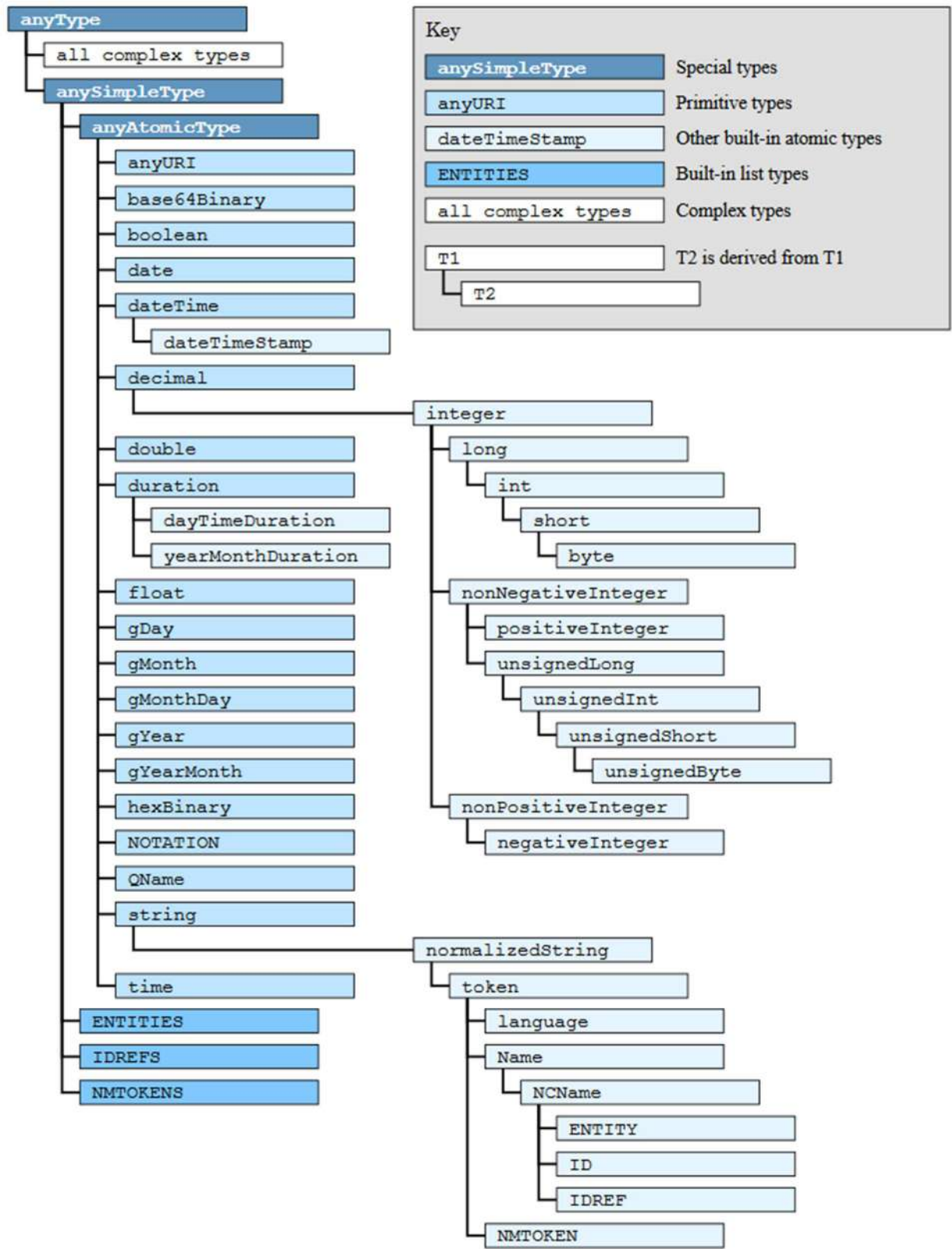
#### 3.5.14.1 Tipos básicos predefinidos

Los tipos predefinidos utilizados para definir el metamodelo utilizar los nombres y la semántica de definición de esquemas XML (XSD) <sup>5</sup>. Además del tipo "LangString" con la semántica definidas en el Marco de Descripción de Recursos (RDF) <sup>6</sup> se utiliza. "LangString" es una cadena que se puede proporcionar en varios idiomas, cada uno.

<sup>5</sup> ver: <https://www.w3.org/XML/Schema>

<sup>6</sup> ver: <https://www.w3.org/TR/rdf11-concepts/>

Figura 28 tipos integrados de [Definición de Esquema XML 1.1 \(XSD\)](#)



## 3.5.14.2 tipos

Tabla 5 tipos básicos utilizados en Metamodel

| Tipo             | Tipo básico   |
|------------------|---------------|
| PropertyValue    | cuerda        |
| QualifierType    | cuerda        |
| Código           | cuerda        |
| anySimpleTypeDef | cuerda        |
| BlobType         | byte [0 .. *] |
| PathType         | cuerda        |

## 3.5.15 Plantillas, herencia, calificadores y Categorías

En una primera vista parece que hay cierta superposición entre el concepto de plantillas de especificación de datos, la herencia, los calificadores y categorías. En esta cláusula las similitudes y diferencias se explican y se dan consejos para las buenas prácticas.

En la extensión general de la metamodelo por herencia está permitido. Como alternativa se podrían utilizar también las plantillas.

- Las plantillas sólo se deben usar si diferentes instancias de la clase siguen diferentes esquemas y las plantillas para los esquemas no se conocen en tiempo de diseño. Las plantillas también pueden ser utilizados si el metamodelo en general todavía no es lo suficientemente estable o una herramienta que hace las plantillas de apoyo, pero (todavía) el metamodelo completa.
- Sin embargo: cuando se utiliza no estandarizado patentada especificación de datos de plantillas de interoperabilidad no puede garantizarse y por lo tanto se debe evitar siempre que sea posible.
- En caso de que todas las instancias de una clase siguen el mismo esquema a continuación, la herencia y / o categorías deben ser utilizados.
- Las categorías pueden ser utilizados si todas las instancias de una clase seguir el mismo esquema, pero tienen diferentes limitaciones en función de su categoría. una restricción de este tipo podría especificar que un atributo opcional es obligatorio para esta categoría (como por ejemplo la unidad que es obligatorio para propiedades que representan valores físicos). Al darse cuenta de la misma a través de la herencia llevaría a la herencia múltiple es lo que se omite.
- Calificadores se utilizan si la semántica del elemento es el mismo independientemente de sus calificadores. Es sólo la calidad o el significado del valor para el elemento que difiere.

### 3.6 plantillas predefinidas de especificación de datos

#### 3.6.1 Concepto de plantillas de especificación de datos

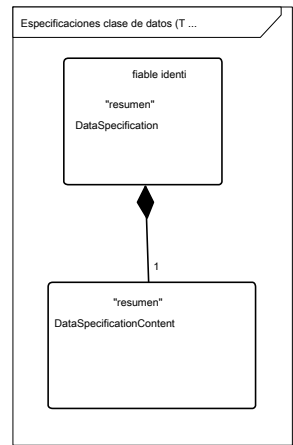


Figura 29 Concepto de plantillas de especificación de datos

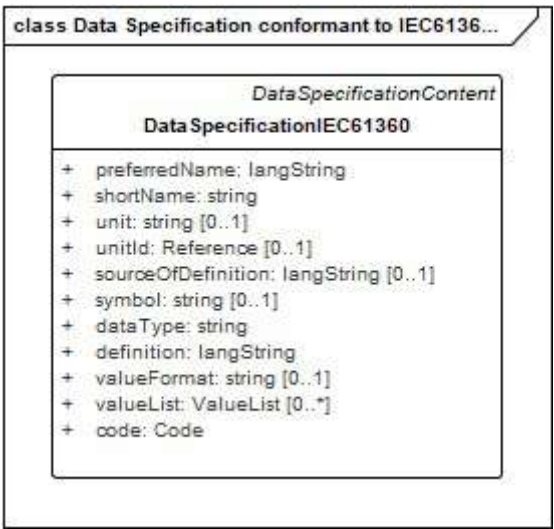
Nota: El plantillas de especificación de datos no pertenecen a la metamodelo de la Concha de Administración de Activos. En serializaciones que eligen plantillas específicas del correspondiente contenido de especificación de datos puede incorporar directamente.

Se requiere que una plantilla de especificación de datos tiene un identificador único global, por lo que se puede hacer referencia a través HasDataSpecification.

Una plantilla se compone de la DataSpecificationContent que contiene los atributos adicionales que se añaden a la instancia de elemento que hace referencia a la plantilla de especificación de datos y metainformación acerca de la propia plantilla (esta es la razón por ejemplar DataSpecification hereda de identificable). En UML estos son dos clases separadas.

#### 3.6.2 Plantillas predefinidas para Descripciones de las propiedades

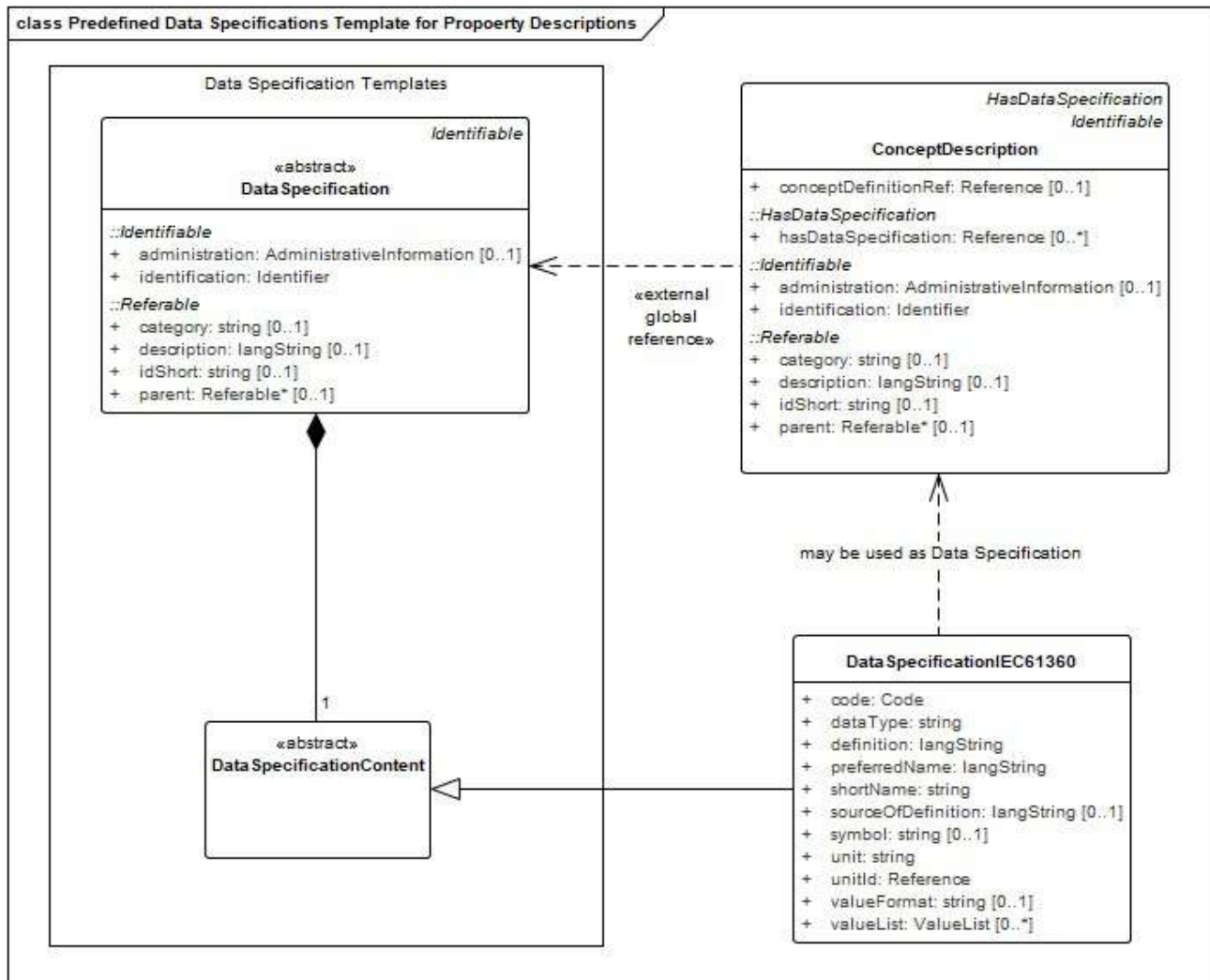
Figura 30 Especificación de la plantilla de datos para definir Descripciones de las propiedades conformes a la norma IEC 61360



Para el significado de los atributos de la plantilla de especificación de datos IEC61360 por favor refiérase a la norma IEC 61360 y / o eCl @ ss. Recomendamos para referirse a esta plantilla de especificación de datos a través de la identificación " [www.adminshell.io/DataSpecificationTemplates/DataSpecificationIEC61360](http://www.adminshell.io/DataSpecificationTemplates/DataSpecificationIEC61360) "(En hasDataSpecification).

Vea la Figura 31 para la forma de plantillas de especificación de datos están relacionados con descripciones de conceptos (mostrando todos los atributos heredados también). En unas plantillas de forma similar para otros elementos en el modelo de información puede ser definido y usado.

Figura 31 Vista general descripciones de concepto y plantillas de especificación de datos



## 4 Asignaciones a los formatos de datos para compartir I4.0- información compatible

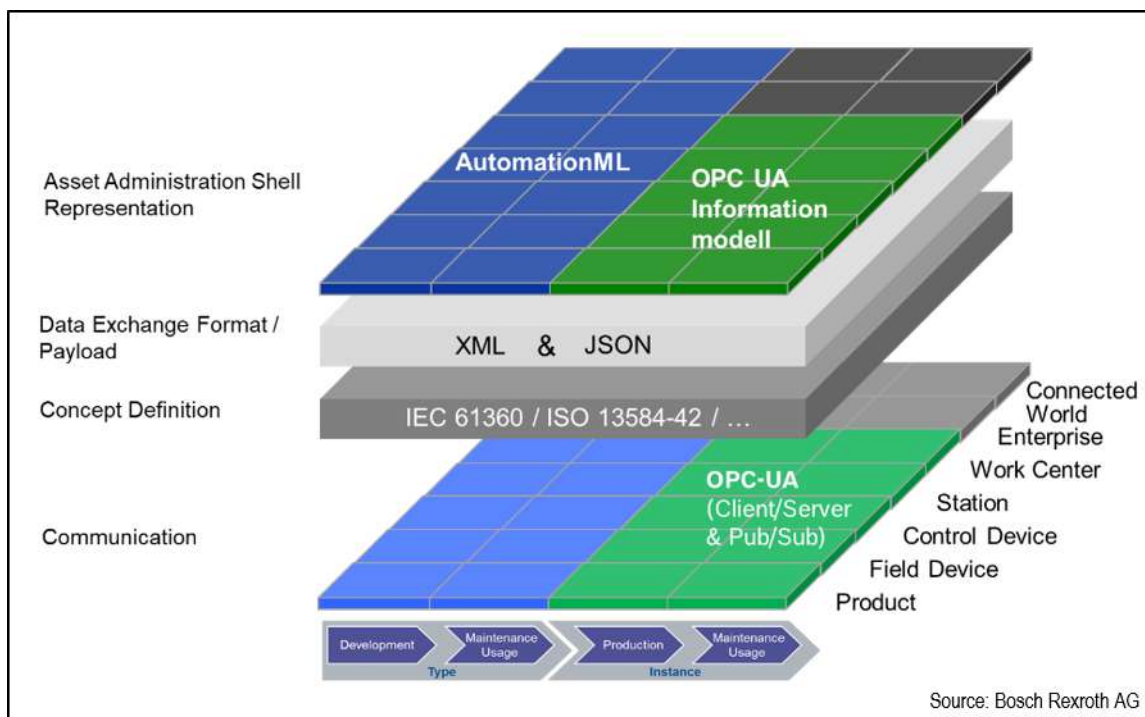
## 4.1 General

Debería ser posible compartir información I4.0 compatible entre diferentes sistemas en todo el área cubierta por todo el modelo RAMI4.0 [1] [2]. OPC UA se ha apuntado como formato para los modelos de información en el ámbito de las operaciones de producción, pero hay una necesidad de otros formatos para las otras áreas y las interrelaciones entre ellos. Este documento describe el Shell AssetAdministration junto con sus submodelos en diferentes formatos de datos <sup>7</sup>:

Tabla 6 Distinción de diferente formato de datos para el AAS

| Formato de datos           | Propósito / motivación   |
|----------------------------|--|
| modelos OPC UA Información | El acceso a toda la información de los datos de la administración y el intercambio de datos en tiempo real dentro de las operaciones de producción. El acceso para los sistemas de fábrica de más alto nivel a esta información. |
| AutomationML               | El intercambio de tipo e instancia de información sobre los activos, en particular durante la ingeniería. La transferencia de esta información en la fase operativa (cf. OPC UA y la asignación correspondiente)                 |
| XML, JSON                  | Serialización de esta información para el propósito de la comunicación técnica entre fases.  |
| RDF                        | Mapeo de esta información que permita un uso completo de las ventajas de las tecnologías semánticas.   |

Figura 32 Vista gráfica en el intercambio de datos Formatos para la administración de activos de Shell <sup>8</sup>



Las especificaciones de la cláusula anterior serán transferidas específicamente a los formatos de intercambio de datos individuales.

## 4.2 Reglas generales

<sup>7</sup> El uso abreviada de la palabra "formatos de datos" incluye el uso de ventajas conceptuales tales como los modelos de información, esquemas, protocolos de transmisión, etc.

<sup>8</sup> Sólo los formatos de datos considerados en este documento se mencionan hasta ahora en la figura.

En la siguiente distinguimos entre las teclas globales y modelo. Ellos se definen como sigue:

- **UNA teclas global es una llave con idtype <> IdShort. Una clave global puede ser local (local = true) si se hace referencia a un elemento dentro de la misma AAS, por ejemplo un ConceptDescription u otro submodelo.**
- **UNA clave de modelo es una clave con el tipo <> GlobalReference, es decir, se hace referencia a un elemento del modelo dentro de los mismos AAS (local = true) o dentro de otros AAS (locales = false).**
- Una distinción similar se realiza para las referencias:
- **UNA modelo de referencia es una cadena de clave de referencia en el que la última tecla es una tecla de modelo.**
- **UNA referencia global es una cadena de clave de referencia en el que la última clave es una clave global con type = GlobalReference.**
- **Un referencia global externa es una referencia global para la cual la primera clave en la cadena de clave de referencia no es local (local = false).**
- **UNA referencia global local es una referencia global para la cual la primera clave en la cadena de clave de referencia es local (local = true).**

Las siguientes reglas se mantienen y aseguran que los posibles referencias cíclicas pueden ser serializado:

- En una cadena de clave de referencia, una clave con local "verdadero" es seguido ya sea por ninguna tecla o una tecla con "local" es "verdadero".
- En una cadena de clave de referencia, una clave con local "falsa" es seguido ya sea por ninguna tecla o una tecla con "local" es "verdadero"

## 4.3 ejemplo unificada

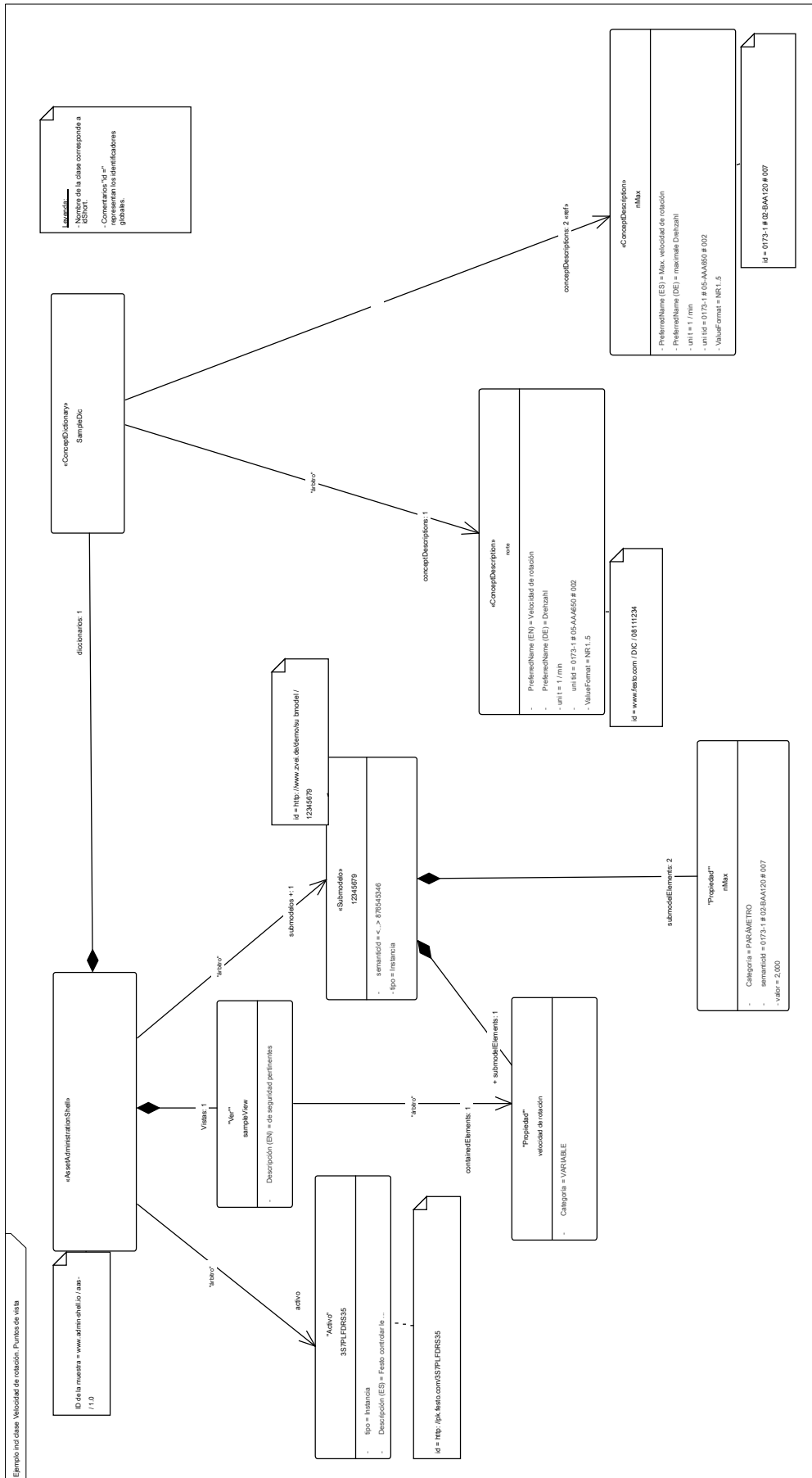
El siguiente ejemplo se utiliza para demostrar las características principales de los formatos de datos como se explica en las siguientes cláusulas para diferentes formatos de datos. La intención es motivar a la equivalencia de información en diferentes representaciones. Los ejemplos mismas se pueden encontrar en el anexo.

Se muestra un AAS con dos propiedades: la velocidad de rotación real (idShort = "rotationSpeed"), un valor de medición (categoría = VARIABLE), así como la velocidad máxima de rotación "Nmax" (categoría = parámetro). El AAS representa un controlador con corta id "3S7PLFDRS35".

Hasta saber que no hay propiedad definida dentro eCI @ ss para la velocidad de rotación real. Por lo tanto se añade una descripción correspondiente concepto (con idShort ="N") al diccionario local de la AAS. Se pone el mundial "id = www.festo.com / DIC / 08111234" identificador que se hace referencia mediante semanticId en la propiedad "rotationSpeed". Para la máxima velocidad de rotación eCI @ ss proporciona una definición semántica con identificador global "# 0173-1 02BAA120 # 007". Una copia de la entrada, se crea en el diccionario local. El id de la copia es el mismo que en eCI @ ss. La unidad física de las propiedades de velocidad de rotación y descripción concepto es 1 / min, denotado por una IRDI único global "0173-1 # 05-AA650 # 002" para 1 / min como se define por eCI @ ss.



Figura 33 Ejemplo velocidad de rotación para la serialización de los formatos de datos



## 4.4 XML

### 4.4.1 General

En las siguientes cláusulas se presenta un resumen de los principales conceptos de la serialización XML AssetAdministration Shell. Para los escenarios de importación y exportación del metamodelo de un Shell AssetAdministration necesita ser serializado. Un formato de serialización es XML. La información se divide en tres partes. La primera parte discute las reglas, en la segunda parte son ejemplos de algunas reglas específicas y en la tercera parte del esquema y un ejemplo completo se muestra en el anexo.

### 4.4.2 Introducción

Extensible Markup Language (XML <sup>9</sup>) es muy adecuado para conseguir la información de un sistema informático, tal vez para procesar de forma manual, y después de darle de comer en otro sistema informático. Por lo tanto, satisface las necesidades del escenario de intercambio de información definido en la Sección 0. XML proporciona para las posibilidades de definiciones de esquema que se pueden utilizar para validar sintácticamente la información representada en cada paso. Por esta razón, este documento proporciona definiciones básicas del esquema para permitir una validación de la información que se comparte. Las definiciones XML se dividen en dos archivos diferentes:

- IEC61360 definición de tipo de datos: iec61360.xsd
- definiciones básicas para la Shell AssetAdministration y su contenedor de exportación: aas.xsd Posteriormente, se

proporciona un ejemplo en el XML.

### 4.4.3 Reglas

Los principales conceptos del esquema XML y la serialización XML resultante se explican por las siguientes reglas. Reglas 1 a 6 son reglas generales, mientras que las reglas 7 a 11 son específicos de la Bibliografía.

#### (1) XSD Tipos globales se utilizan para el modelado

Para XSD reutilización tipos globales serán utilizados para el modelado. Habrá una convención de nomenclatura  
`<InformationModelName> + '_' + 't'`

#### (2) Si está presente, los nombres se toman del modelo de información.

Por razones de comprensibilidad los nombres clave XML debe ser el mismo que el elemento que representa en el metamodelo.

#### (3) Todos identificables tienen una agregación a nivel de la raíz.

Los identificables son AssetAdministrationShells, activos, submodelos, ConceptDescriptions. Para reducir los casos de redundancia, que se encuentran exclusivamente en la agregación de nivel superior.

#### (4) la repetición de elementos y sus tipos recibirán los mismos nombres de sus casos en plural.

Si el elemento tiene una cardinalidad de  $n > 1$  de un elemento primario será utilizado con el nombre del nombre del elemento en plural. Por ejemplo, **cada elemento en la agregación bienes tiene que ser un activo.**

#### (5) Identificables que no están en las agregaciones de nivel superior son sólo referencias a los correspondientes

instancias en una de las agregaciones de nivel superior.

Esta regla se completa el concepto de la regla 3. No debería haber ninguna identificación redundante en el metamodelo serializado.

#### (6) Para los elementos con el tipo LangString se añade un elemento de agregación. Para el elemento único una lengua

se añade etiqueta "lang".

Para los propósitos de internacionalización esta regla es necesario.

#### (7) Los atributos de una clave en una referencia excepto por el valor en sí se realizan como atributos XML.

#### (8) Plantillas Especificación Los datos se añade directamente a la Descripción del concepto, ya que hasta ahora sólo

descripciones de las propiedades son compatibles.

---

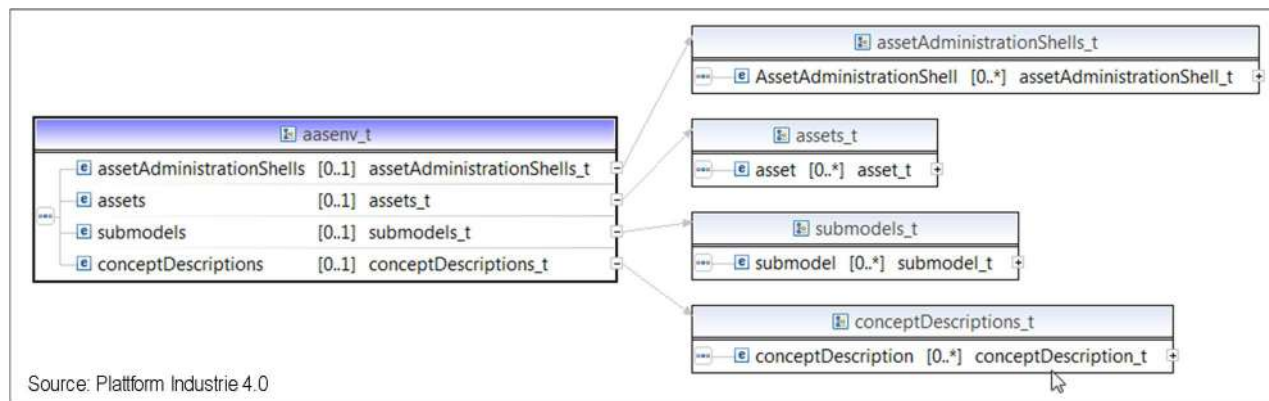
<sup>9</sup> ver: [https://de.wikipedia.org/wiki/Extensible\\_Markup\\_Language](https://de.wikipedia.org/wiki/Extensible_Markup_Language)

Además, un nuevo EmbeddedDataSpecification elemento es introducido que tiene dos atributos: una para la referencia global para el identificador de especificación de datos y otro para el contenido de la especificación de datos.

#### 4.4.4 Ejemplo de estructuras de nivel superior

Una serialización describe un entorno de Shell de administración de activos que es una colección de conchas de administración. El elemento raíz del medio ambiente AssetAdministration Shell tiene 4 agregaciones. Para cada clase de identificación, uno de agregación se presenta, como exige la regla 3.

Figura 34 estructura de nivel superior de un entorno AssetAdministration Shell correlacionarse con el esquema XML



Nota: XSD estructuración se hizo con cadena de herramientas de Eclipse

El XML resultante es el XML mínimo:

Tabla 7 Minimal XML para la estructura de nivel superior

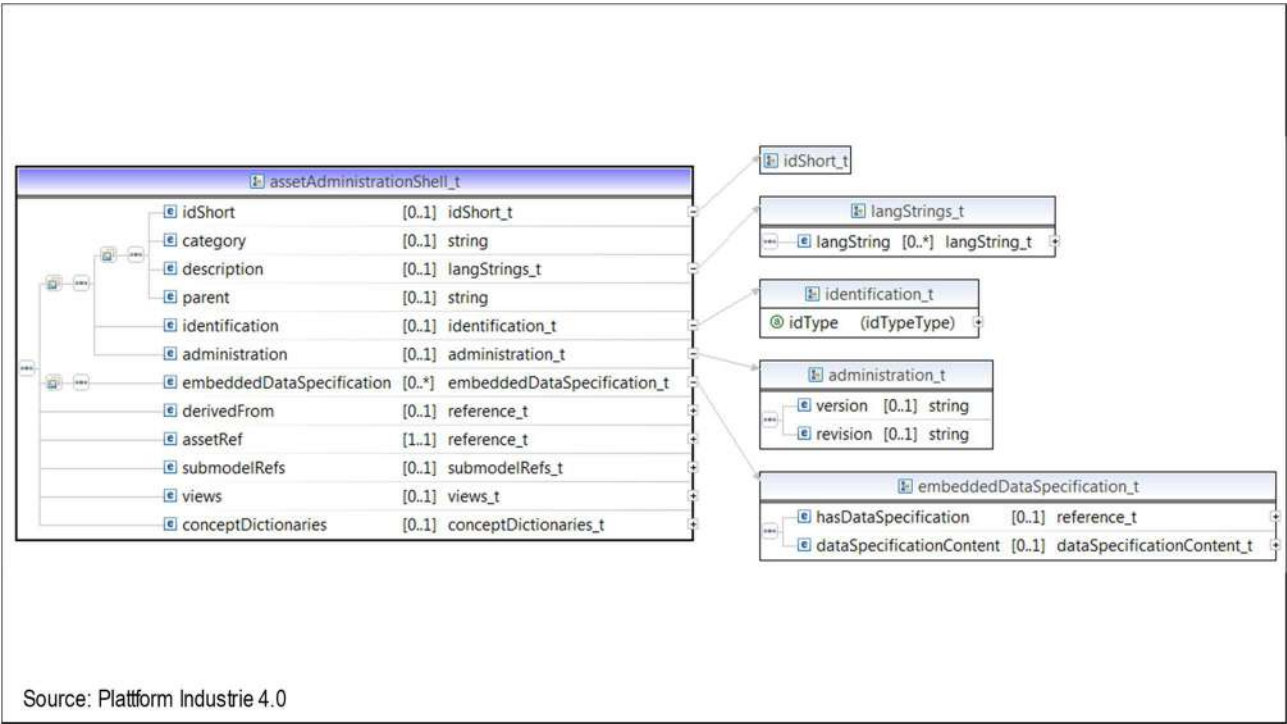
|  |
|--|
| <pre> &lt;? xml versión = "1.0" codificación = "UTF-8" ?&gt; &lt;Aas: aasenv xmlns: AAS = "Http://www.admin-shell.io/aas/1/0" •   xmlns: xsi = "Http://www.w3.org/2001/XMLSchema-instance" •   xsi: schemaLocation = "Http://www.admin-shell.io/aas/1/0 AAS.xsd" &gt; &lt;Aas: assetAdministrationShells&gt; &lt;/ aas: assetAdministrationShells&gt; &lt;aas: activos&gt; &lt;/ aas: activos&gt; &lt;aas: submodelos&gt; &lt;/ aas: submodelos&gt;    &lt;aas: conceptDescriptions&gt; &lt;/ aas: conceptDescriptions&gt; &lt;/ aas: aaenv&gt; </pre> |
|--|

Nota: • designa de ajuste de línea para fines de diseño

4.4.5 XSD grupos de modelos

Hay una serie de grupos de atributos en el modelo UML - es decir, identificable o hasSemantics. Estos grupos se modelan como grupos de modelos XSD para que pudieran ser reutilizados como grupos anónimos en diferentes lugares.

Figura 35 XSD grupos de modelos



Esto se realiza en el XSD según como sigue:

Tabla 8 Uso de grupos de modelos XSD

```
<complexType nombre = "AssetAdministrationShell_t" > <Secuencia>

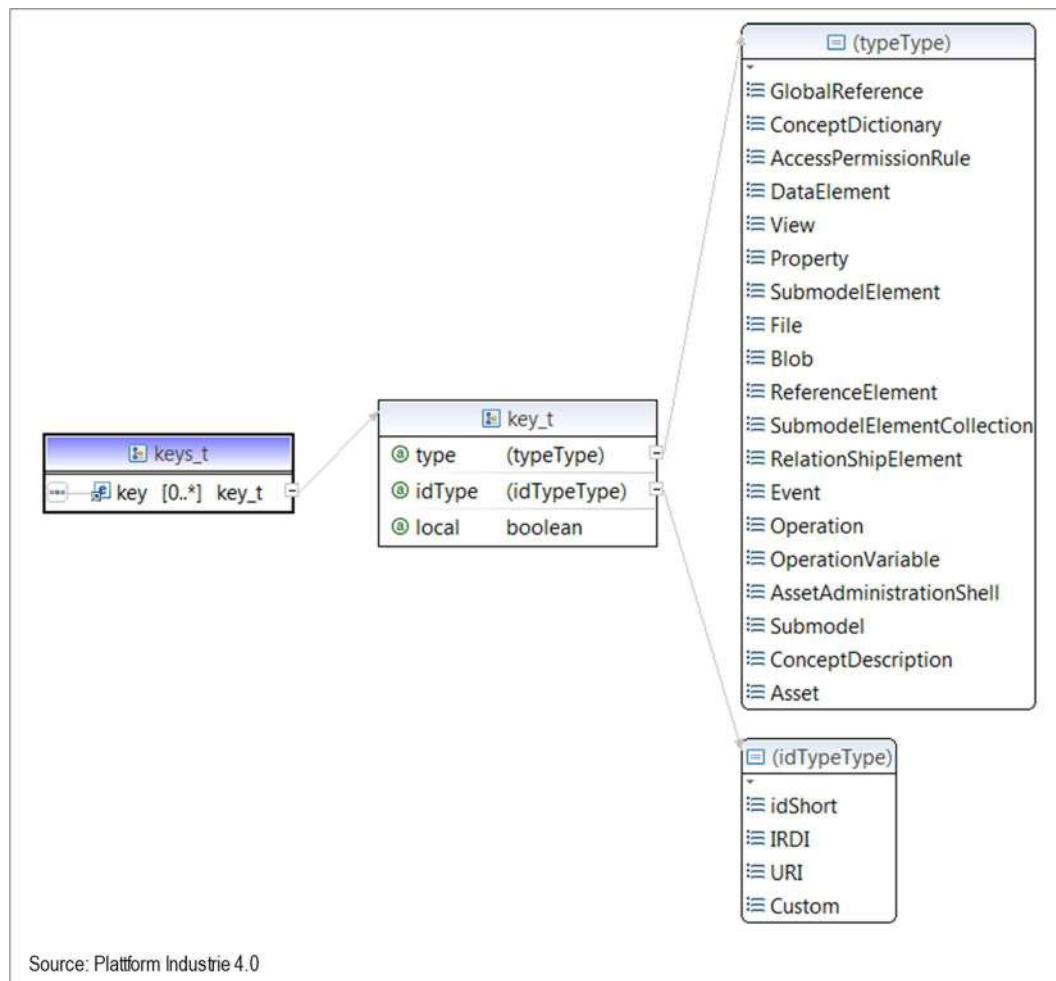
    <grupo árbitro = "Aas: identificable" > </ Grupo> <grupo árbitro = "Aas: hasDataSpecification" > </ Grupo> <elemento nombre = "derivado
de" tipo = "Aas: reference_t" /> <Elemento nombre = "AssetRef" tipo = "Aas: reference_t" /> <Elemento nombre = "submodelRefs" tipo = "Aas:
submodelsRef_t" /> <Elemento nombre = "puntos de vista" tipo = "Aas: viewsRef_t" > <Elemento nombre = "conceptDictionaries" tipo = "Aas:
conceptDictionaries_t" /> </ Secuencia> </ complexType>
```

Nota: debido al mecanismo de grupos XSD, mapas hasDataSpecification a un elemento de embeddedDataSpecification\_t y mapas identificables a varios elementos de la figura 35.

#### 4.4.6 Teclas y Referencias

Las claves y referencias (véase 3.5.13) se asignan en la misma construcción de esquemas XML. Algunos de los atributos definidos tienen enumeraciones - éstos se asignan en las restricciones de cuerda.

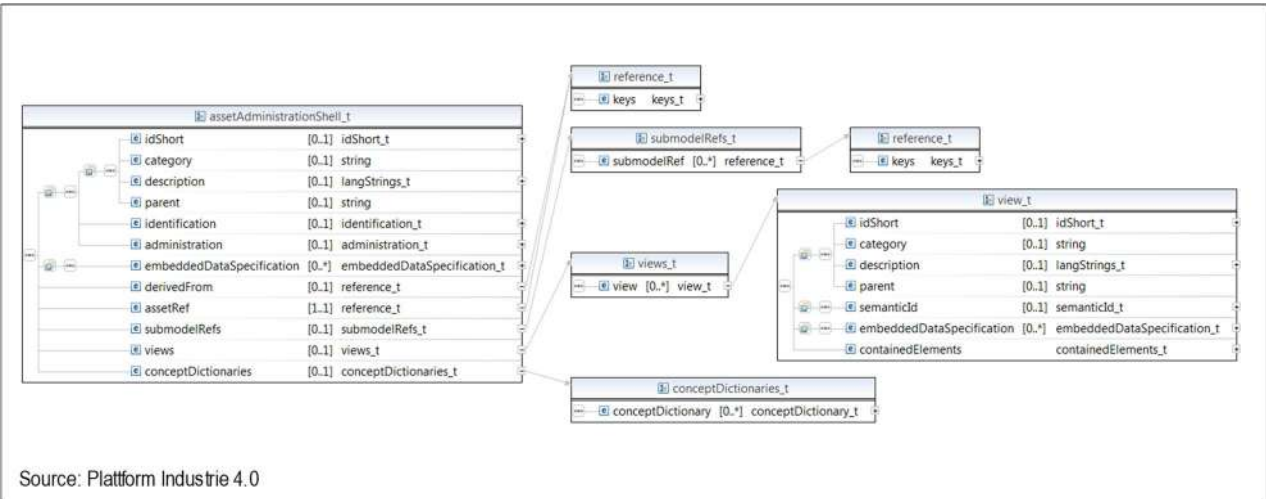
Figura 36 Teclas y Referencias



#### 4.4.7 Mapeo de Shell de administración de activos

Los depósitos de la Administración de activos se asignan siguiendo el siguiente esquema XML - consiste en un conjunto de parámetros de metadatos y sobre todo enlaces a otras partes del documento XML o a entidades externas (basados en claves y referencias).

Figura 37 Vista general en la cartografía y meta-datos

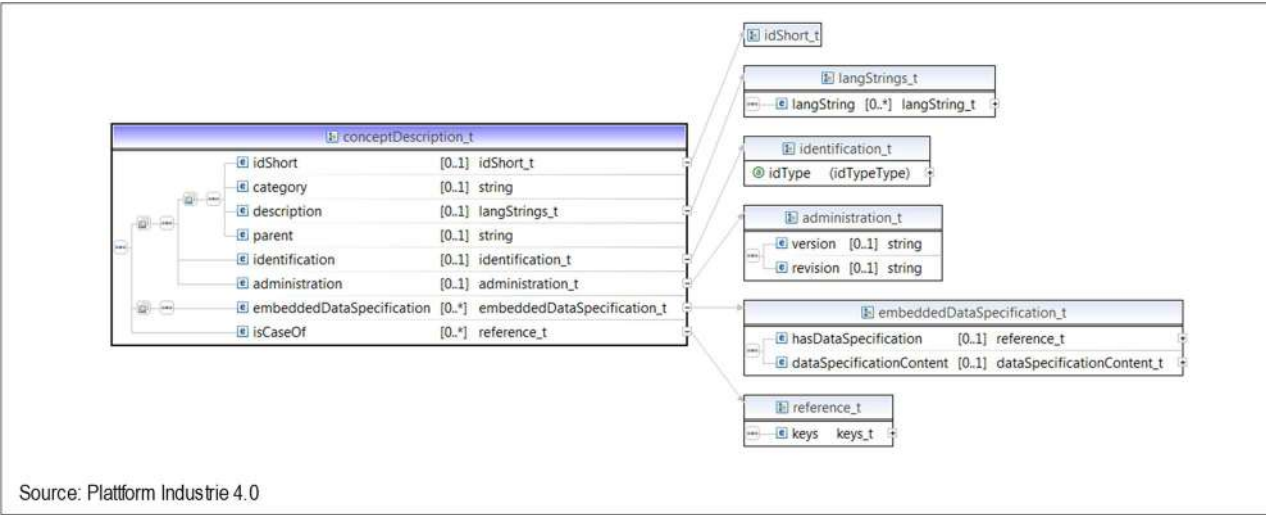


4.4.8 ConceptDescriptions y EmbeddedDataSpecifications Mapeo

Como se describió anteriormente, la definición de un concepto comprende de una referencia de acuerdo y un contenido, que se realiza por una especificación de datos.

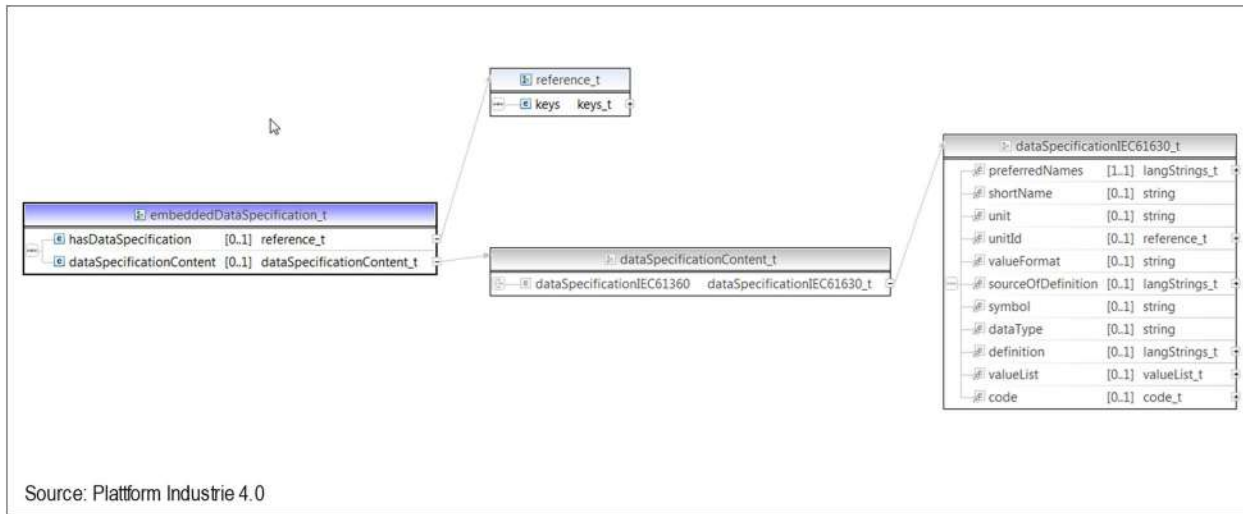
Nota: EmbeddedDataSpecification fue nombrado "ConceptDefinition" en una antigua versión del metamodelo. los cifras aún no se actualizan.

Figura 38 Descripción Concept en XML en general



La especificación de datos puede ser, por ejemplo a lo largo de un IEC 61360 propiedad:

Figura 39 especificación de datos a través de la norma IEC 61360 atributos de propiedad



XSD completo y el ejemplo XML se pueden encontrar en el Anexo D.

## 4.5 JSON

### 4.5.1 General

En las siguientes cláusulas se presenta un resumen de los principales conceptos de la serialización JSON AssetAdministration Shell. Para los escenarios de importación y exportación del metamodelo de un Shell AssetAdministration necesita ser serializado. Un formato de serialización es JSON <sup>10</sup> (JavaScript Object Notation). La información se divide en tres partes. La primera parte discute las reglas, en la segunda parte son ejemplos de algunas reglas específicas y en la tercera parte del esquema y un ejemplo completo se muestra en el anexo.

El proyecto financiado con fondos públicos BaSys 4.0 proporciona una implementación de código abierto de la Shell de administración de activos y su serialización JSON a finales de 2018. <sup>11</sup>

### 4.5.2 Reglas

Los principales conceptos de la serialización JSON se explican por las 11 reglas siguientes. Reglas 1 a 6 son reglas generales, mientras que las reglas 7 a 11 son específicos de la Bibliografía.

**(1) Si está presente, los nombres se toman del modelo de información.**

Por razones de comprensibilidad los nombres clave JSON debe ser el mismo que el elemento que representa en el metamodelo.

**(2) Cada objeto tiene un atributo adicional "modelType" con el nombre de la clase de objeto correspondiente**

como valor

Esta norma es necesaria por razones de deserialización.

**(3) Todos identificables tienen una agregación a nivel de la raíz.**

Los identificables son AssetAdministrationShells, activos, y submodelos ConceptDescriptions. Para reducir los casos de redundancia, que se encuentran exclusivamente en la agregación de nivel superior.

**(4) Nombres de agregación son los nombres de sus casos en plural.**

Si el valor de un par de valores clave es una matriz JSON el nombre de clave debe ser el nombre de las instancias de esta matriz JSON en plural.

Por ejemplo, cada objeto en los activos de agregación tiene que ser un activo.

**(5) Identificables que no están en las agregaciones de nivel superior son sólo referencias a los correspondientes**

instancias en una de las agregaciones de nivel superior.

Esta regla se completa el concepto de la regla 3. No debería haber ninguna identificación redundante en el metamodelo serializado.

**(6) El (en varios idiomas) Descripción en el metamodelo es siempre una agregación de las descripciones de la JSON serializado.**

Para los propósitos de internacionalización esta regla es necesario.

**(7) Todas las colecciones ordenadas incluyendo teclas tienen un índice. El primer objeto de la colección tiene el índice 0.**

Debido a que la cadena de clave de referencia es una lista ordenada se necesita el atributo de índice.

**(8) Plantillas Especificación Los datos se añade directamente a la Descripción del concepto, ya que hasta ahora sólo**

descripciones de las propiedades son compatibles.

Además, un nuevo EmbeddedDataSpecification elemento es introducido que tiene dos atributos: una para la referencia global para el identificador de especificación de datos y otro para el contenido de la especificación de datos.

### 4.5.3 Ejemplo de estructuras de nivel superior

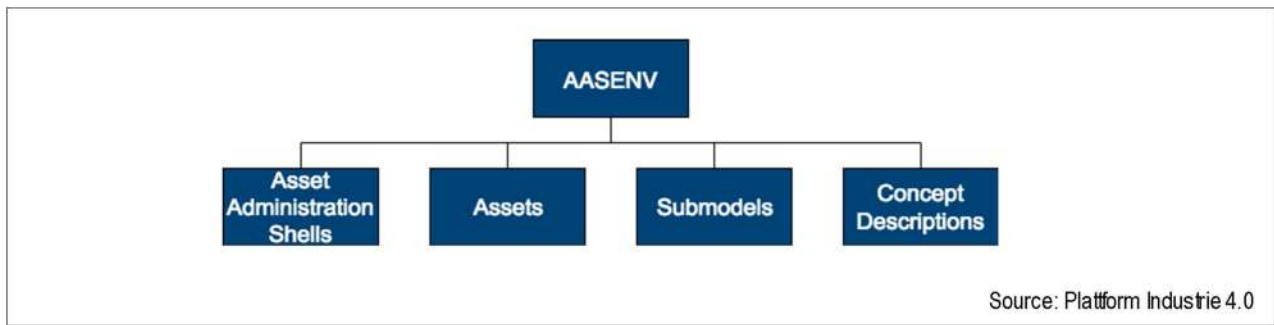
Una serialización describe un activo entorno de Shell Administración, es decir, una colección de conchas de administración. El elemento raíz del medio ambiente AssetAdministration Shell tiene 4 agregaciones. Para cada clase de identificación personal, una agregación es características, como exige la regla 3.

<sup>10</sup> ver: <https://tools.ietf.org/html/rfc8259> o <https://www.ecma-international.org/publications/standards/Ecma-404.htm>

<sup>11</sup> BaSys 4.0 SDK implementación de código abierto, véase: <https://projects.eclipse.org/projects/technology.basysx>



Figura 40 estructura de nivel superior de un entorno AssetAdministration Shell asigna a JSON



El JSON resultante es el JSON válido mínimo:

Tabla 9 JSON Minimal para la estructura de alto nivel

```

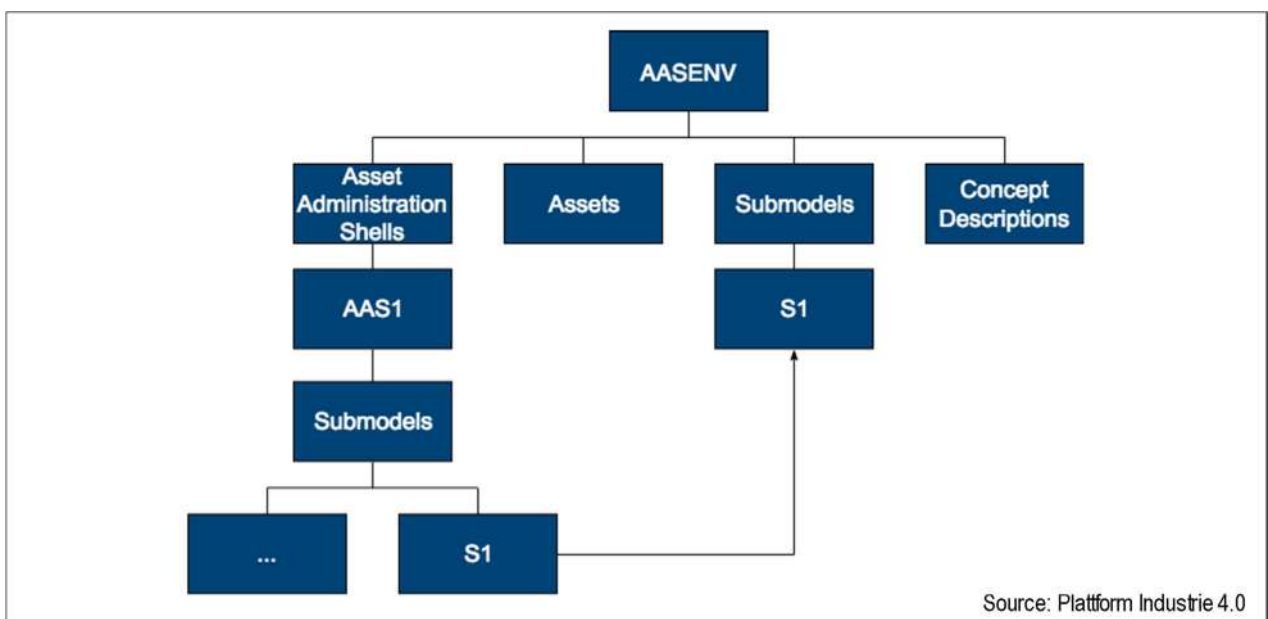
{
  "assetAdministrationShells" : [],
  "bienes" : [],
  "submodelos" : [],
  "conceptDescriptions" : []
}
  
```

#### 4.5.4 Ejemplos de referencias a iables IDENTIF

Como exige la regla 5, identificables sólo se les permite ubicarse en las agregaciones de alto nivel. En las partes más profundas de la estructura sólo referencias a la correspondiente toma lugar identificable.

En el AssetAdministration Shell AAS1, el submodelo S1 es sólo una referencia a la instancia Submodelo S1 en la agregación el nivel superior submodelos.

Figura 41 referencia Submodelo en AssetAdministrationShell para JSON



Esto se traduce en la siguiente JSON ejemplar:

Tabla 10 JSON mínimo ejemplar para Referencias

|  |   |
|--|---|
| <pre>{   "assetAdministrationShells": [     {       "tipo de modelo": "AssetAdministrationShell",       "submodelos": [         {           "llaves": [             {               "KeyType": "URI",               "local": cierto,               "tipo": "Submodelo",               "valor": "Http://env.com/submodels/S1",               "indice": 0             }           ]         }       ]     }   ],   ... }, "bienes": []</pre> | <pre>"submodelos": [   {     "tipo de modelo": "Submodelo",     "identificación": {       "carné de identidad": "Http://env.com/submodels/S1",       "tipo de identificación": "URI"     },     "IdShort": "S1",     "submodelElements": [],     ...   } ], "conceptDescriptions": []</pre> |
|--|---|

4.5.5 Ejemplos para las descripciones

Como se describe en la regla 6, una descripción de la serialización es una matriz de las descripciones de la metamodelo.

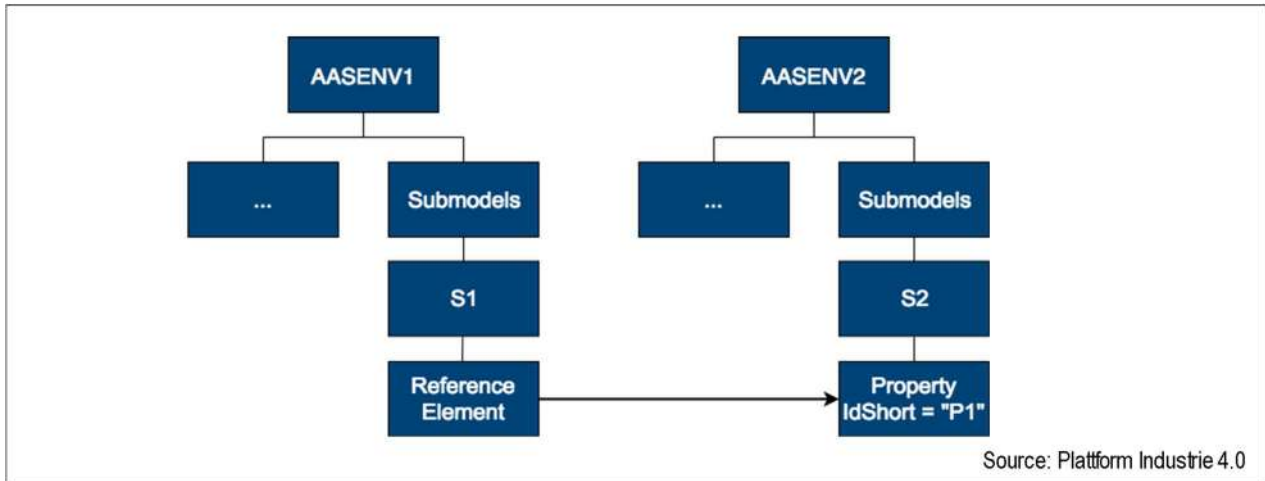
Tabla 11 JSON mínimo Ejemplar para la estructura de alto nivel

|  |
|--|
| <pre>"descripciones": [   {     "texto": "Beispiel Beschreibung",     "idioma": "DELAWARE"   }, {     "texto": "Descripción de la muestra",     "idioma": "ES"   } ]</pre> |
|--|

4.5.6 Ejemplos para ReferenceElement

A ReferenceElement tiene una referencia como valor. Esta referencia tiene una agregación de teclas que representa un llavero. Los resueltos puntos clave de la cadena a un elemento. En este ejemplo puntos de valor de la ReferenceElement a una propiedad de otro submodelo en otro ambiente Shell de administración de activos. La primera clave es una clave global con el atributo "local" se pone a falso, es decir, la referencia no es parte del propio entorno. La segunda clave es una clave de modelo que se utiliza para definir la propiedad correspondiente en el otro entorno por su IdShort. Es una buena práctica utilizar el llavero más corto si hay varias opciones.

Figura 42 Uso de ReferenceElement en JSON



Esto resulta en un JSON ejemplar como tal:

Tabla 12 ReferenceElement ejemplar en JSON

```

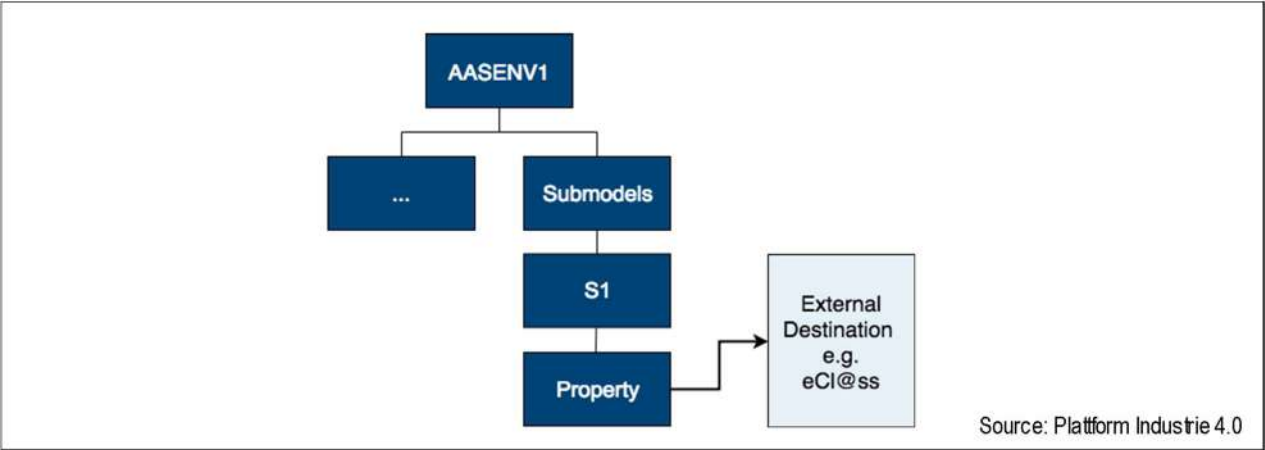
{
  "llaves" : [
    {
      "KeyType": "URI",
      "Local": false,
      "tipo": "Submodelo",
      "valor": "Http://admin-shell.io/submodels/Temperature",
      "indice": 0
    }, {
      "KeyType": "IdShort",
      "Local": true,
      "tipo": "Propiedad",
      "valor": "NMax",
      "indice": 1
    }
  ]
}

```

4.5.7 Ejemplos para GlobalReference

A veces es útil hacer referencia a otro estándar o algo que no es proporcionado por el propio entorno AssetAdministration Shell. En este ejemplo, la semántica de una propiedad se refiere a eCI @ ss.

Figura 43 Uso de GlobalReference en JSON



Esto resulta en un JSON ejemplar como tal:

Tabla 13 GlobalReference ejemplar en JSON

```
{
  "llaves" : [
    {
      "KeyType": "IRDI", "local": false,

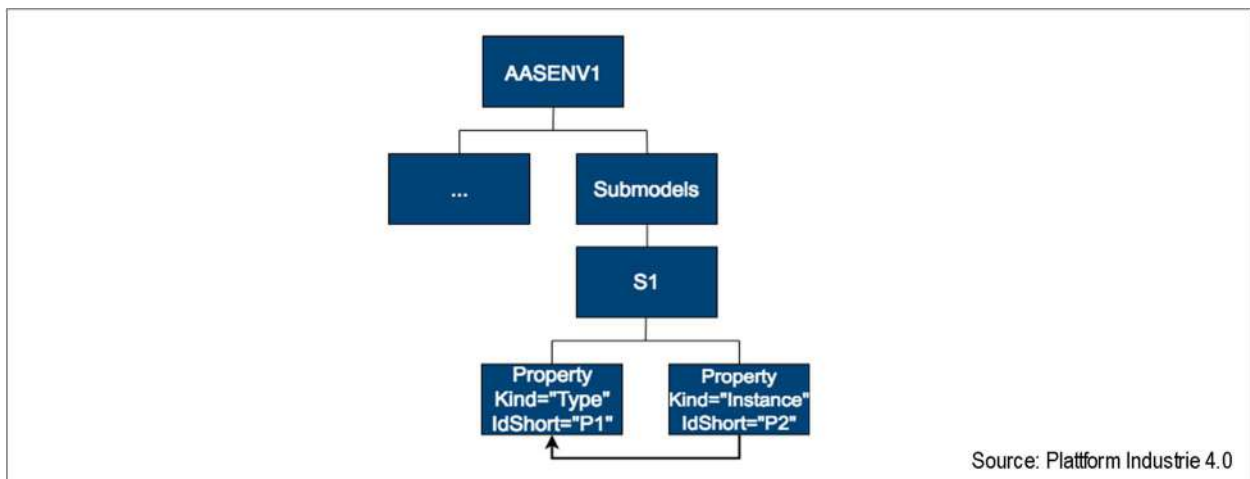
      "tipo": "GlobalReference",
      "valor": "0173 - 1 # 02 - AAC962 # 006 ",
      "indice": 0
    }
  ]
}
```

#### 4.5.8 Ejemplo para un tipo = "tipo" Referencia

Una descripción semántica puede ser algo externo o una instancia con el tipo = "tipo". En este ejemplo, la propiedad P2 P1 utiliza como plantilla. P1 tiene tipo = "tipo" y P2 tipo = "ejemplo".

**Nota:** típicamente, los tipos se supone que se especifica en otro Shell Administración de activos como las instancias. Aquí, la situación descrita se simplifica por razones de diseño.

Figura 44 Ejemplar tipo de referencia en JSON



Esto resulta en un JSON ejemplar como tal:

Tabla 14 Ejemplar tipo de referencia en JSON

```

{
  "llaves" : [
    {
      "KeyType" : "URI", "local": true,
      "tipo" : "Submodelo",
      "valor" : "http://aasenv1.com/submodel/S1",
      "indice" : 0
    }, {
      "KeyType" : "IdShort", "local": true,
      "tipo" : "Propiedad",
      "valor" : "P1",
      "indice" : 1
    }
  ]
}

```

## 4.6 RDF

En noviembre de 2018, el mapeo hacia RDF está en discusión. Los resultados estarán disponibles tan pronto como estén finalizados.

## 4.7 OPC UA

Las obras de la asignación a la OPC Unified Architecture se llevan a cabo actualmente en un grupo de trabajo conjunto <sup>12</sup> entre OPC Fundación, ZVEI y VDMA. Los resultados estarán disponibles tan pronto como estén finalizados.

## 4.8 AutomationML

A partir de noviembre de 2018, la asignación hacia AutomationML están actualmente en proceso de trabajo. Los resultados estarán disponibles tan pronto como estén finalizados.

---

<sup>12</sup> ver: <https://opcfoundation.org/collaboration/i4aas/>



5 Atributo de acceso basado en roles y la base



## 5.1 Permisos de Acceso pasando

Al tener un vistazo a la imagen principal (Figura 1 en la cláusula 2.2) también aspectos de seguridad tienen que ser considerados cuando la transferencia de información de socio de la cadena un valor al siguiente.

Cuando el contenido de administración shell se pasa de una pareja a la siguiente, los siguientes pasos se deben hacer, aquí se muestra para el ejemplo de que el proveedor pasa sobre el contenido de al integrador:

- Paso A1-A2: El proveedor hace una elección que hay datos a ser transmitida (véase la cláusula 5.2), y por lo tanto determina el contenido del paquete AASX (véase la cláusula 6).
- Paso A2-A3: El paquete AASX se transfiere al integrador.
- Paso A3-A4: El integrador recibe el paquete e importa el contenido en su dominio de seguridad. Durante este paso, el integrador tiene que establecer los derechos de acceso de acuerdo con los requisitos de su propio dominio de seguridad. Esto demuestra que los derechos de acceso son independientes entre los dos dominios de seguridad.

La cáscara de administración utiliza control de acceso basado atributo (ABAC), un papel puede ser considerado como un atributo en este contexto; otros atributos podrían ser la hora del día, la dirección de origen y otros. Dos condiciones de contorno requieren la transmisión de los permisos de acceso entre los socios:

**(una)** Los permisos de acceso a los elementos de información de un AAS deben establecerse en cada dominio de seguridad.

**(segundo)** Una pareja debe ser capaz de pasar una sugerencia que se deben establecer permisos de acceso para el activo que se describe en el AAS.

Un ejemplo para el segundo requisito: un fabricante robot sugiere que para el robot deben establecerse las siguientes funciones: setter máquina, el operador y un papel de mantenimiento. También sugiere permisos para estas funciones, por ejemplo, un instalador tiene acceso de escritura en el programa del robot, pero el operador no lo hace.

El ejemplo anterior motiva, que la semántica de las reglas de permiso de acceso y sus definiciones exactas que se deben transmitir de un dominio de seguridad al otro.

La transmisión de la semántica de acceso basado atributo es implementado por los siguientes medios:

- Definición de permisos de acceso: El permiso de acceso detallado (por ejemplo, leer, escribir, borrar, crear, invocar el método etc.) se definen en un **sub-modelo de dominio específico (ver defaultPermissions y selectablePermissions en la cláusula 5.4.5).**
- Definición de las reglas de permiso de acceso, basado en los permisos de acceso definidos. Estos se definen como parte de control de acceso (véase la cláusula 5.4.6).
- Asociación de reglas de permiso de acceso a cada elemento de información (objeto) de la AAS. Esto significa es realizado por la estructura de **información de la AAS, en sí (ver PermissionsPerObject en la cláusula 5.4.6).** En [19] ejemplos y más información general sobre el control de acceso de atributo y el control de acceso, en general, se puede encontrar.

### 5.1.1 Acceso efectivo basado en reglas de permiso de acceso

permisos de acceso eficaces se determinan en base a las reglas de permiso de acceso.

Cada elemento de información (objeto) en el AAS tendrá reglas que define sus permisos de acceso para cada sujeto. El sujeto se asume que ya está autenticado.

Si un elemento de información no tiene estas reglas, se utilizará automáticamente la tabla para el elemento en el que se incluye ( "herencia desde arriba"). El objeto más superior es el propio AAS, es decir, el AAS es el punto de partida para la herencia .

Como se ha indicado antes, identificación del sujeto, la regla definiciones y también permisos podría ser diferente para el dominio de seguridad de recepción.

Cuando la parte receptora establece permisos de acceso durante el paso A3-A4, debe establecer el pasado-en las definiciones de acceso (permisos y reglas de permiso de acceso) a las definiciones existentes en su dominio de seguridad.

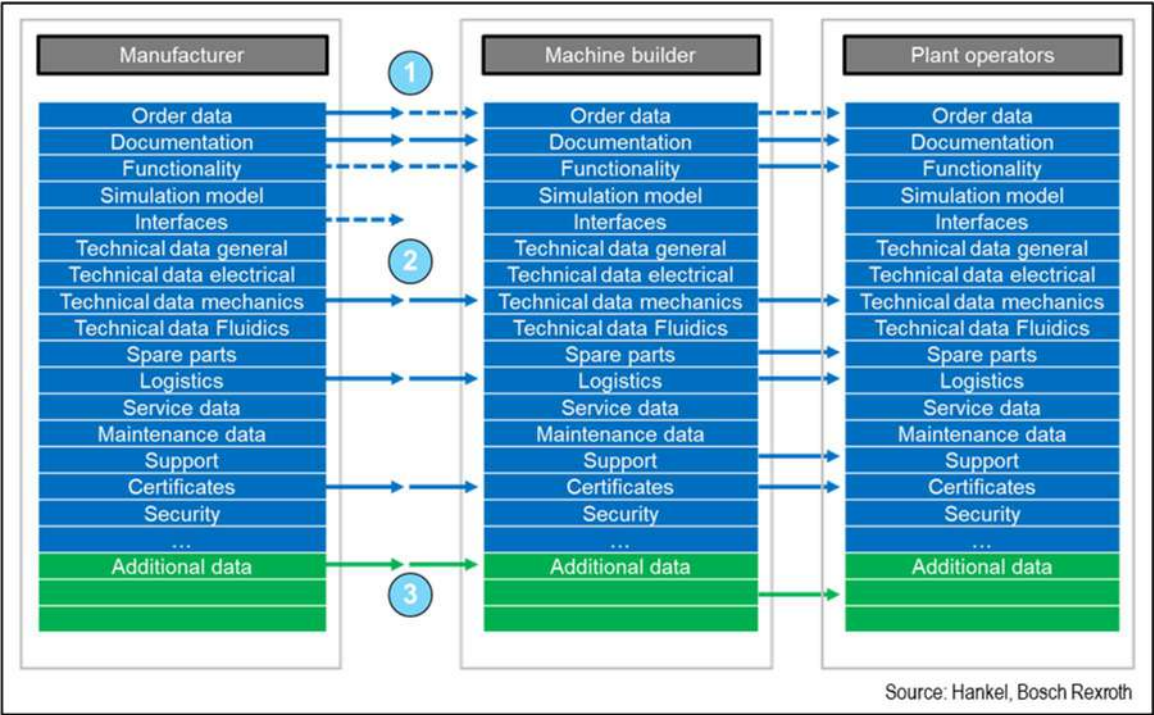
También cuenta con un mapa de la regla del permiso de acceso recibida por elemento de información a los mecanismos de permisos existentes en su dominio de seguridad.

5.2 Filtrado de la Información en exportación e importación

Cuando el intercambio de información de los socios de A a B pareja hay dos casos de uso:

- El productor de la información no quiere enviar la información completa, sino sólo partes de él. La información presentada puede variar dependiendo del consumo específico de la información es presentada al. Es decir, se necesita un mecanismo de filtrado que permite dar forma a la información individual para el consumidor específico.
- El consumidor de la información no quiere incluir toda la información proporcionada por el productor de la información en su propio proceso, es decir, que quiere filtrar sólo la información relevante.

Figura 45 Ejemplo de filtrado de exportación e importación

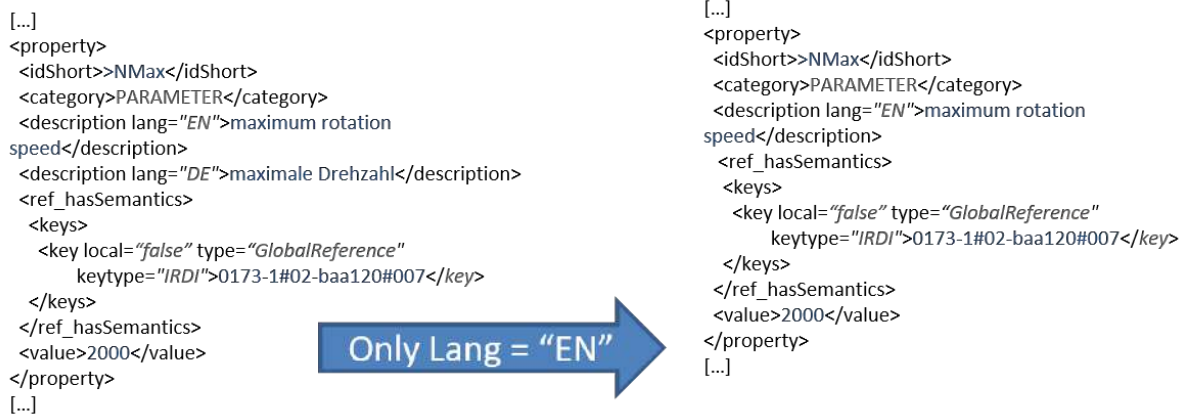


Como un ejemplo, supongamos que el productor es la presentación de los datos de pedido completas. Sin embargo, el consumidor (en este caso, los fabricantes de maquinaria) es el filtrado de la información (1) y sólo está importando la información relevante para él. Para la funcionalidad tanto a filtrar: el productor está filtrando lo que somete al consumidor (2) y el consumidor una vez más no está utilizando todas las funciones, pero está filtrando de nuevo, que la funcionalidad se utiliza en su entorno. Lo mismo es posible entre los fabricantes de maquinaria y operador.

Nota: En el caso de uso considerados en este documento, el intercambio de información a través de intercambio de archivos XML, etc. El información que no está destinada a ser presentada tiene que ser extraído de los archivos XML correspondientes antes de la entrega o antes de la importación, respectivamente. Papel o atributos de control de acceso no encajan aquí. Las políticas de acceso correspondientes podrían ayudar a filtrar la información correspondiente, pero que no pueden ser presentados como parte del archivo correspondiente intercambiado.

La Tabla 15 muestra un ejemplo cuando se utiliza el formato XML se define tal como se define en este documento. En el ejemplo no se presentará la traducción al alemán, se proporciona único idioma Inglés para el socio B.

Tabla 15 Ejemplo de filtrado de la información en XML



### 5.3 Shell Resumen de la administración de activos metamodelo para la Seguridad

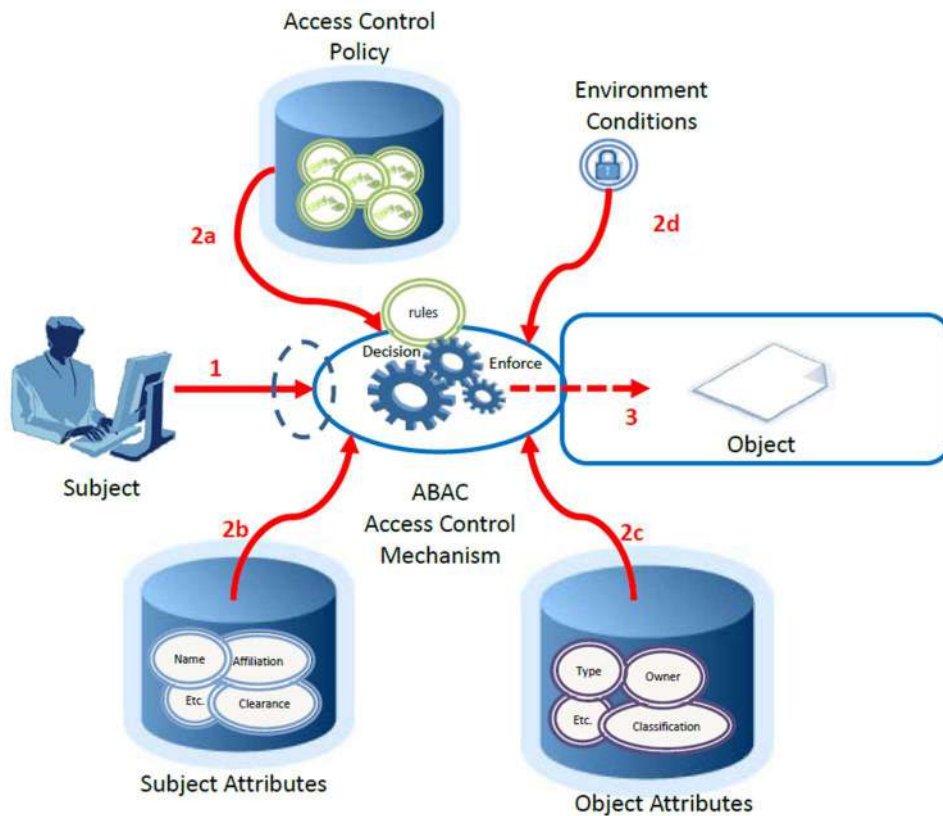
Los atributos de seguridad son una parte obligatoria de cualquier Administración de Shell. Los atributos de

seguridad describen:

- Puntos de política de control de acceso, incluyendo definición de reglas de permiso de acceso
- anclas de confianza

En este documento, principalmente el aspecto de permiso de acceso se trata. El concepto subyacente es el concepto de control de atributo basado acceso (ABAC) como se describe en [22].

Figura 46 Atributo de control de acceso basado en [22]



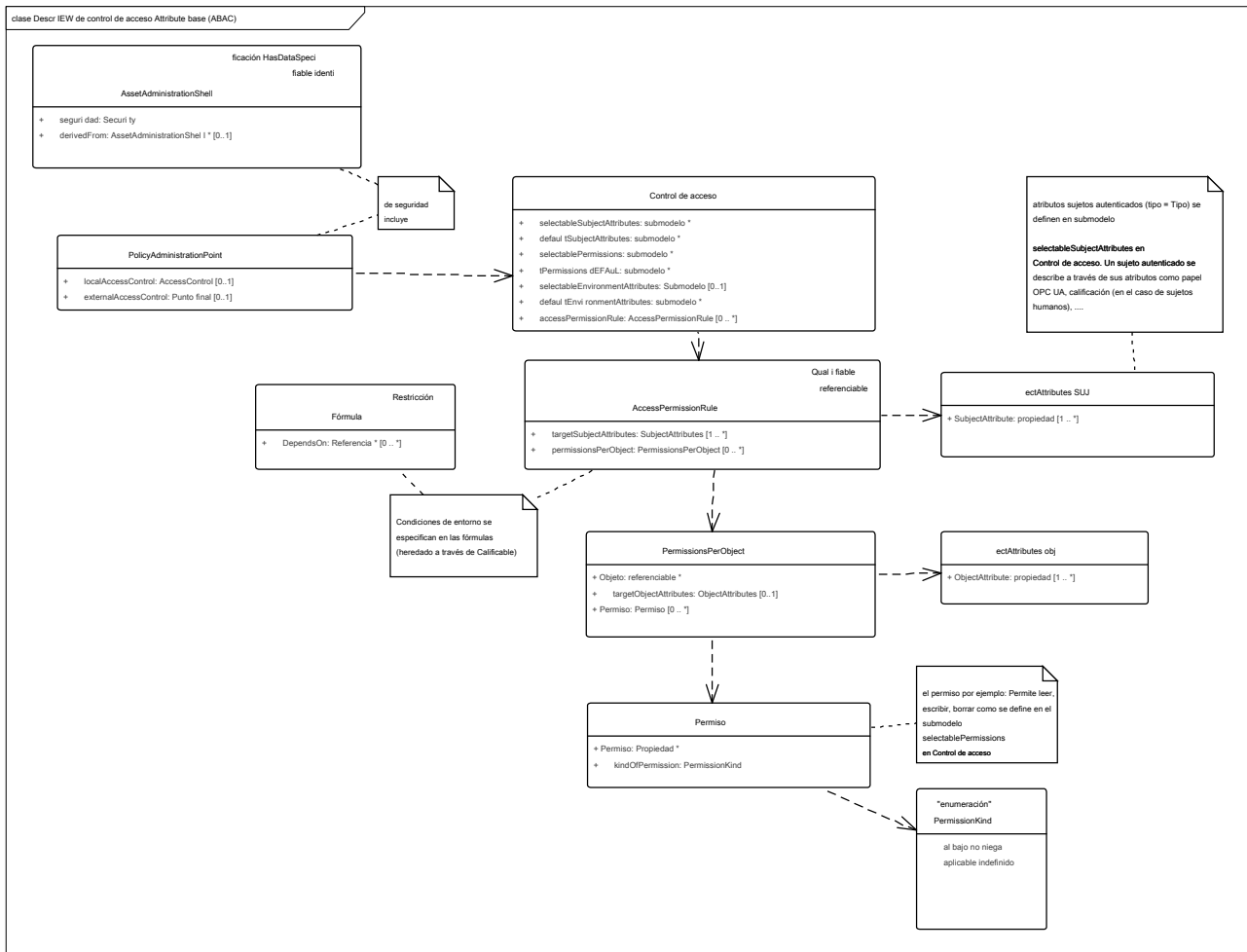
Nota: Atributo en el contexto de ABAC es diferente de atributos de los elementos definidos en el metamodelo.

El concepto general se representa en la Figura 46: Un sujeto está solicitando acceso a un objeto (1). En el contexto de un AAS un objeto típicamente es un submodelo o una propiedad o cualquier otro elemento submodelo conectado al activo. El mecanismo de control de acceso en práctica de la AAS evalúa las reglas de permiso de acceso (2a) que incluyen limitaciones que deben ser cumplidas wrt los atributos de sujeto (2b), el objeto atributos (2c) y las condiciones de entorno (2d). En la Figura 47 se da una visión general del modelo de información de la seguridad wrt AAS. La atención se centra en el control de acceso. Un objeto en el contexto de ABAC corresponde típicamente a un submodelo o a un elemento de submodelo. El objeto atributos de nuevo se modelan como elementos submodelos.

Atributos Asunto necesitan ser visitada, ya sea a través de un punto de información política exterior o que se definen como las propiedades dentro de un submodelo especial de la AAS. Un atributo típico sujeto es su papel. El papel es el único atributo de objeto definido en caso de control de acceso basado en roles.

Opcionalmente, las condiciones del entorno pueden ser definidos. En el control de acceso basado en roles no se han definido las condiciones del ambiente. Condiciones ambientales pueden ser expresados a través de restricciones de fórmula. Para ser capaz de hacer lo que los valores necesarios deberían definirse como la propiedad o referencia a datos dentro de un submodelo de la AAS.

Figura 47 Metamodel Descripción general del control de acceso de AAS



A través de reglas de permiso de acceso se define qué sujeto se le permite acceder qué objetos <sup>13</sup> dentro del AAS. Se supone que el tema ya está autenticado. Los objetos pueden ser cualquier elemento atribuible, es decir, incluyen submodelos, activos, descripciones de conceptos, puntos de vista, etc. Más en general, se puede especificar si se permite o se niega a acceder a un objeto un sujeto autenticado aso "acceso" podría ser uno de los permisos especificados en un elemento de la AAS. ¿Qué permisos son seleccionables no se define por el metamodelo de la AAS. Los permisos seleccionables se definen a través de un submodelo ( **selectablePermissions**). Lo mismo vale para los atributos sujetos ( **selectableSubjectAttributes**). atribuye el asunto por defecto y se utilizan permisos predeterminados si no son sobrescritos por el dueño de la AAS. En cuanto a permisos de los atributos sujetos utilizado autenticados se definen en submodelo **selectableSubjectAttributes**.

A través de las restricciones fórmula los derechos de acceso podrían ser aún más limitada. Por ejemplo, una fórmula podría especificar que el papel "ingeniero de mantenimiento" (para ser más precisos: un sujeto autenticado con objeto atributo "papel = 'ingeniero de mantenimiento'") sólo se permite escribir parámetros de configuración si la máquina (el activo) no es corriendo. Vea la Figura 15 en la cláusula 3.5.2.6 de la expresión formal de esta regla de acceso basado en la propiedad "Estado".

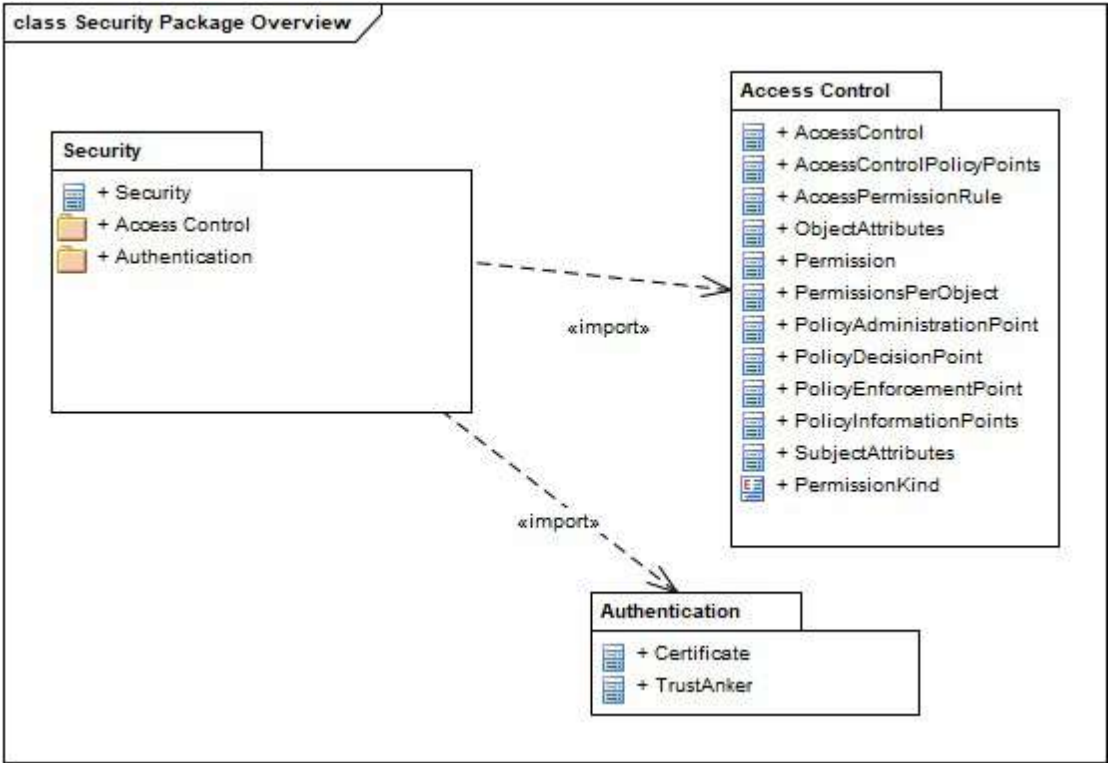
Atributos de los objetos se manejan de una manera diferente. Se supone que cualquier propiedad del objeto en foco puede tener, además, sobre el papel de un atributo de objeto. Por lo tanto no hay submodelo especial para predeterminado o atributos de los objetos seleccionables. También el control de acceso basado más tradicional papel pueda ser ejercido por un AAS: en este caso no hay limitaciones (= atributos definidos para el medio ambiente) las reglas de control de acceso. Para un sujeto sólo un atributo sujeto necesita ser definida: su papel. Para el objeto sin atributos de objetos adicionales necesitan ser definidos.

<sup>13</sup> El término "objeto" se utiliza porque es más genérico y en el futuro también otros objetos como por ejemplo de los atributos de las clases se pueden incluir, además de los elementos.

Para más detalles sobre el control de acceso basado en atributos, incluyendo ejemplos cómo aplicar el metamodelo como se define en este documento, véase [19].

Las clases y sus atributos se definen en la siguiente cláusula 5.4. La Figura 48 proporciona una visión general de todos los elementos definidos para los problemas de seguridad en el metamodelo.

Figura 48 Paquetes de Introducción a la seguridad



## 5.4 Metamodelo detalles Especificación: Los designadores

### 5.4.1 Introducción

En esta cláusula las clases del metamodelo relacionados con la seguridad se especifican en detalle. Es una extensión del metamodelo como se describe en la cláusula 3.5.

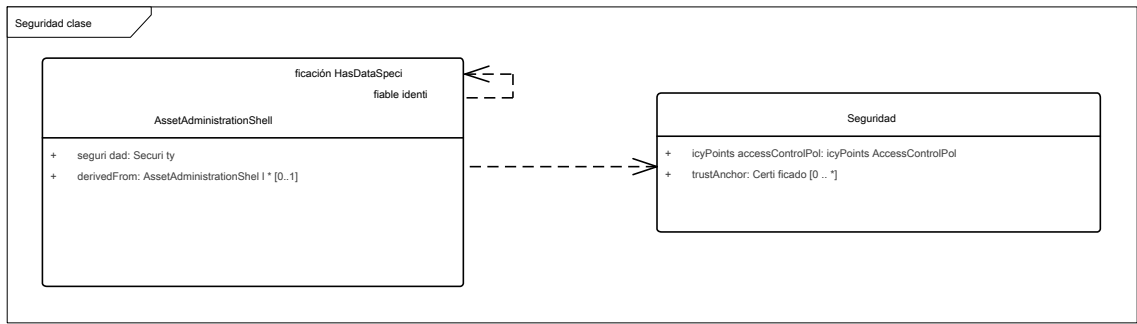
Para comprender la extensión de los conceptos básicos y las clases abstractas comunes deben ser entendidos (véase especialmente la cláusula 3.5.2, 3.5.13 cláusula y la cláusula 3.5.14).

### 5.4.2 Común

punto final sin embargo, no se especifica en detalle en el metamodelo actual. Es sólo una clase abstracta.

5.4.3 Los atributos de seguridad

Figura 49 Metamodel de atributos de seguridad de AAS



|                             |  |                           |      |          |
|-----------------------------|--|---------------------------|------|----------|
| Clase:                      | Seguridad  |                           |      |          |
| Explicación:                | Recipiente para seguridad de la información relevante de la AAS. |                           |      |          |
| Hereda de:                  |  |                           |      |          |
| Atributo (* = obligatorio)  | Explicación  | Tipo                      | Tipo | Tarjeta. |
| accessControlPolicyPoints * | puntos de política de control de acceso de la AAS.               | AccessControlPolicyPoints | aggr | 1        |
| trustAnchor                 | La confianza de anclaje de AAS, típicamente certificados.        | Certificado               | aggr | 0 .. *   |

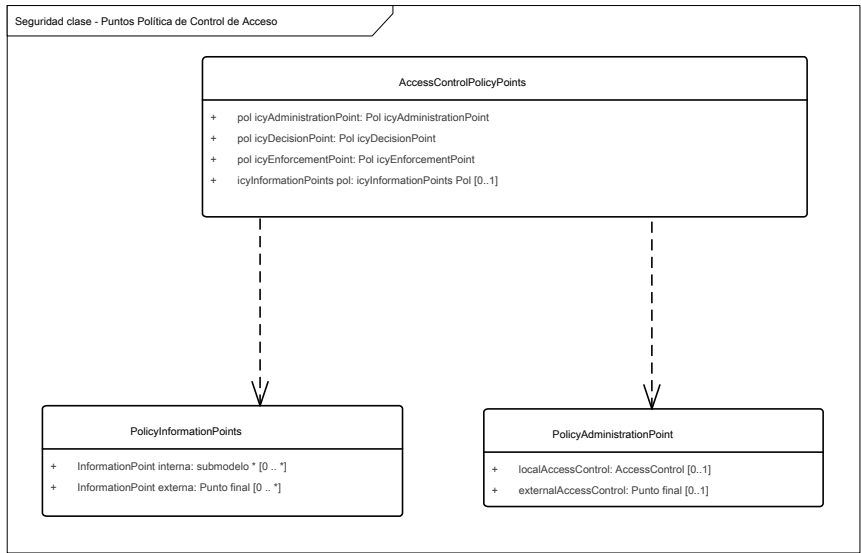
En general, se tiene que considerar cómo habilitar la primera configuración de la seguridad WRT AAS. Esto incluiría el establecimiento del proveedor de la autorización de punto final, etc.

No hay una sola ancla de confianza por AAS ya que los certificados se pueden sobrescribir si un AAS es asumida por un nuevo propietario. El nuevo propietario agrega un nuevo certificado. Sin embargo, el conjunto completo de certificados debe estar disponible.

**Certificado aún no se define más.**

5.4.4 Atributos Política de Control de Punto de Acceso

Figura 50 Metamodel para Control de Acceso



| Clase:                      | AccessControlPolicyPoints   |                           |      |          |
|-----------------------------|---|---------------------------|------|----------|
| Explicación:                | Recipiente para los puntos de política de control de acceso.                  |                           |      |          |
| Hereda de:                  |   |                           |      |          |
| Atributo (* = obligatorio)  | Explicación   | Tipo                      | Tipo | Tarjeta. |
| policyAdministrationPoint * | El punto de la política de la administración del control de acceso de la AAS. | PolicyAdministrationPoint | aggr | 1        |
| policyDecisionPoint *       | El punto de decisión política de control de acceso de la AAS.                 | PolicyDecisionPoint       | aggr | 1        |
| policyEnforcementPoint *    | El punto de aplicación de políticas de control de acceso de la AAS.           | PolicyEnforcementPoint    | aggr | 1        |
| policyInformationPoints     | Los puntos de información política de control de acceso de la AAS.            | PolicyInformationPoints   | aggr | 0..1     |

La definición del punto de decisión de políticas (PDP) es tomado de [22]. El PIP calcula las decisiones de acceso mediante la evaluación de los puntos de decisión y políticas de meta. Una de las principales funciones del punto de decisión política consiste en mediar o deconflict políticas de toma de acuerdo con las políticas del meta. O bien la toma de decisiones se lleva a cabo dentro de la AAS. A continuación, el AAS es autónomo e independiente de un sistema de control de acceso externo. O la toma de decisiones se lleva a cabo fuera de la AAS. A continuación, el AAS tiene que ser capaz de acceder a este criterio de valoración externa para la toma de decisiones.

| Clase:                     | PolicyAdministrationPoint  |                   |                |      |
|----------------------------|--|-------------------|----------------|------|
| Explicación:               | Definición de un punto de administración de la seguridad (PDP).  |                   |                |      |
| Hereda de:                 |  |                   |                |      |
| Atributo (* = obligatorio) | Explicación  | Tipo              | Tarjeta buena. |      |
| localAccessControl         | El punto de decisión política de control de acceso como se dio cuenta por el propio AAS.<br><br><u>Constraint aasd-009: o hay un punto final punto de administración de políticas externo definido o el AAS tiene su propio control de acceso.</u> | Control de acceso | aggr           | 0..1 |
| externalAccessControl      | Punto de llegada a un control de acceso externa que define un punto de administración de políticas para ser utilizado por el AAS.  | punto final       | árbitro*       | 0..1 |

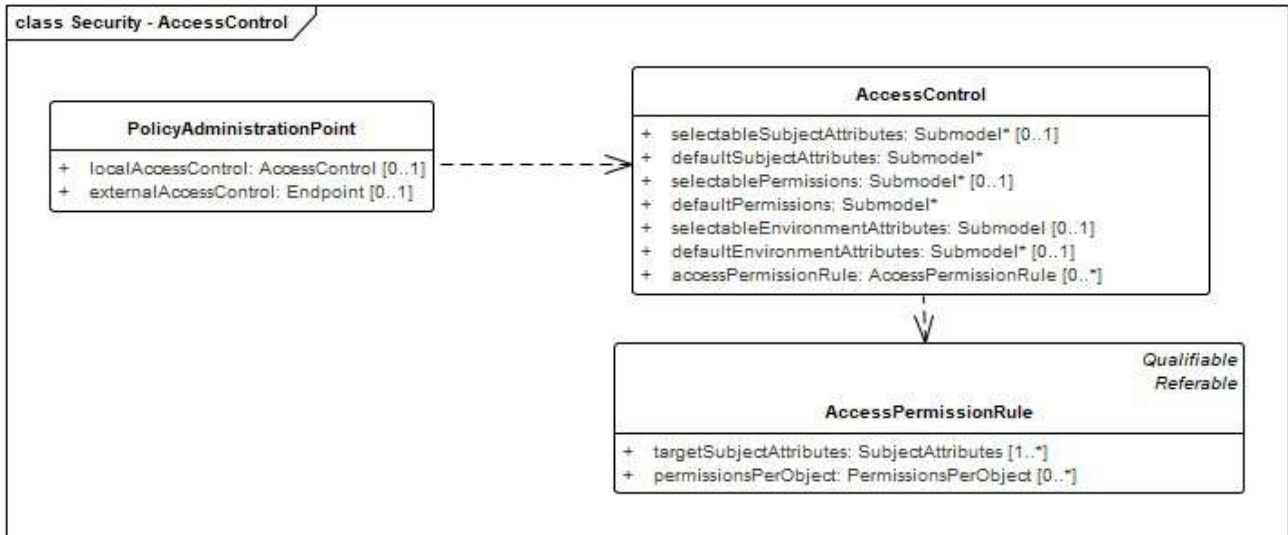
| Clase:                     | PolicyInformationPoints  |             |                |      |
|----------------------------|--|-------------|----------------|------|
| Explicación:               | Define los puntos de información de la política de seguridad (PIP).<br><br>Sirve como fuente de recuperación de atributos, o los datos necesarios para la evaluación de políticas para proporcionar la información que necesita el punto de decisión política para tomar las decisiones. |             |                |      |
| Hereda de:                 |  |             |                |      |
| Atributo (* = obligatorio) | Explicación  | Tipo        | Tarjeta buena. |      |
| externalInformationPoint   | Los puntos finales a la información externa disponible apunta tomando en consideración para el control de acceso para el AAS.  | punto final | aggr           | 0..* |
| internalInformationPoint   | Las referencias a definir submodelos información utilizada por reglas de permiso de acceso de seguridad.   | submodelo   | árbitro*       | 0..* |



La definición de punto de información de política (PIP) se toma de [22]. La diferencia entre los puntos de información externos e internos es si el AAS necesita acceso a través de un punto final a una fuente externa de información o si las tiendas de AAS la información necesaria en sí. También puede haber puntos de información externa e interna de un AAS para ser considerado para la toma de decisiones.

#### 5.4.5 Los atributos de control de acceso locales

Figura 51 Metamodel para Control de Acceso

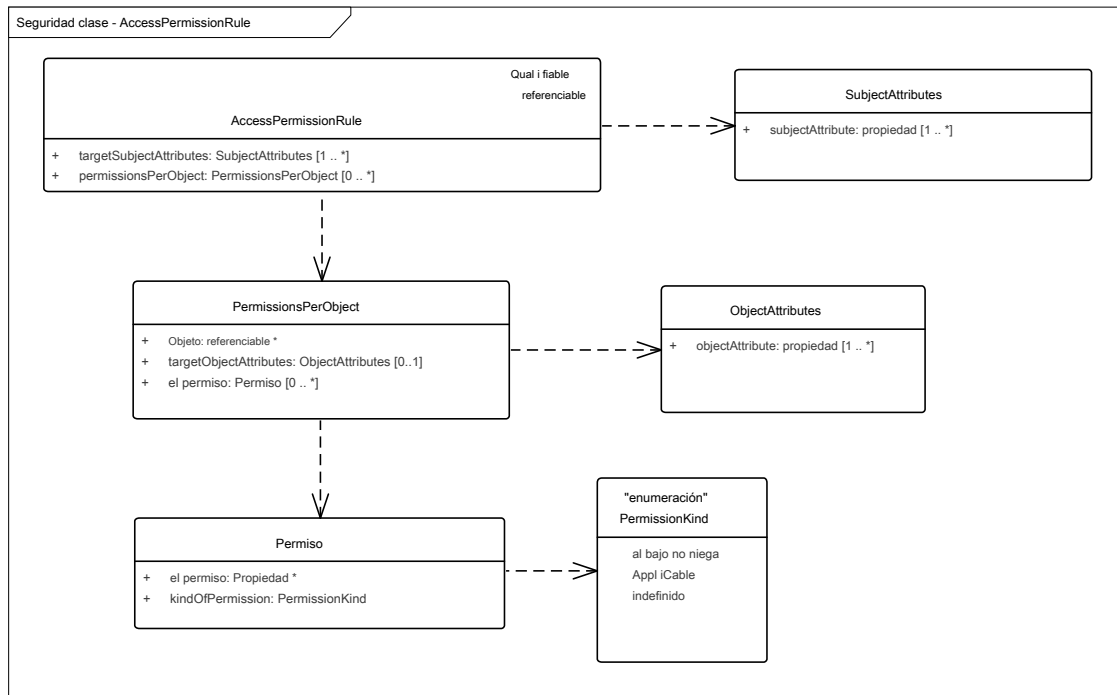


| Clase:                        | Control de acceso  |                          |          |          |
|-------------------------------|--|--------------------------|----------|----------|
| Explicación:                  | Control de acceso define el punto de administración de políticas de control de acceso local. Control de acceso tiene la importante tarea de definir las reglas de permiso de acceso.   |                          |          |          |
| Hereda de:                    |  |                          |          |          |
| Atributo (* = obligatorio)    | Explicación  | Tipo                     | Tipo     | Tarjeta. |
| accessPermissionRule          | reglas de permiso de acceso del AAS que describen los derechos asignados a los sujetos que han sido autenticados () para acceder a los elementos de la AAS.  | AccessPermissi<br>onRule | aggr     | 0 .. *   |
| selectableSubjectAttributes * | La referencia a un submodelo definición de los temas autenticados que están configurados para el AAS. Ellos son seleccionables por las reglas de permiso de acceso para asignar permisos a los sujetos. Por defecto: referencia a la sub-modelo de referencia a través de _____<br>defaultSubjectAttributes. | submodelo                | árbitro* | 0..1     |
| defaultSubjectAttributes *    | La referencia a un submodelo definición de los temas predeterminados de atributos para el AAS que se puede utilizar para describir reglas de permiso de acceso. El submodelo es de tipo = Tipo.  | submodelo                | árbitro* | 1        |
| selectablePermissions *       | La referencia a una definición submodelo que permisos se pueden asignar a los sujetos. Por defecto: referencia a la sub-modelo de referencia a través de _____<br>defaultPermissions   | submodelo                | árbitro* | 0..1     |
| defaultPermissions *          | La referencia a un submodelo la definición de los permisos predeterminados para el AAS.  | submodelo                | árbitro* | 1        |

| Clase:                          | Control de acceso   |           |          |      |
|---------------------------------|---|-----------|----------|------|
| selectableEnvironmentAttributes | <p>La referencia a una definición submodelo que los atributos de entorno se pueden acceder a través de las reglas de permisos definidos para el AAS, es decir, atributos que no están describiendo el propio activo.</p> <p><u>Por defecto:</u> referencia a la sub-modelo de referencia a través de defaultEnvironmentAttributes</p>   | submodelo | árbitro* | 0..1 |
| defaultEnvironmentAttributes    | <p>La referencia a un entorno predeterminado submodelo definir los atributos, es decir, atributos que no están describiendo el activo en sí.</p> <p>El submodelo es de tipo = Tipo. Al mismo tipo de los valores de estos atributos de entorno tienen que ser accesibles al evaluar las reglas de permiso de acceso. Esto se realiza como un punto de información de la política.</p> | submodelo | árbitro* | 0..1 |

## 5.4.6 Los atributos de la regla de permisos de acceso

Figura 52 Metamodel para la Regla permiso de acceso



| Clase:                     | AccessPermissionRule  |                      |      |          |
|----------------------------|---|----------------------|------|----------|
| Explicación:               | Tabla que define permisos de acceso por sujeto autenticado para un conjunto de objetos (elementos referibles).              |                      |      |          |
| Hereda de:                 | Referible; calificable  |                      |      |          |
| Atributo (* = obligatorio) | Explicación   | Tipo                 | Tipo | Tarjeta. |
| targetSubjectAttributes *  | Objetivo sujetos atributos que deben cumplirse por el sujeto que accede para obtener los permisos definidos por esta regla. | SubjectAttributes    | aggr | 1 .. *   |
| permissionsPerObject *     | Conjunto de pares de objetos en permisos que definen los permisos por objeto dentro de la regla de permiso de acceso.       | PermissionsPerObject | aggr | 1 .. *   |

| Clase:                     | PermissionPerObject   |                  |      |          |
|----------------------------|---|------------------|------|----------|
| Explicación:               | Tabla que define permisos de acceso para un objeto especificado. El objeto es cualquier elemento referibles en el AAS. Además los atributos del objeto pueden definirse más que especifican el tipo de objeto de los permisos se aplican a. |                  |      |          |
| Hereda de:                 | --  |                  |      |          |
| Atributo (* = obligatorio) | Explicación   | Tipo             | Tipo | Tarjeta. |
| objeto*                    | Elemento al que se le asigna el permiso.  | Referible        | attr | 1        |
| targetObjectAttributes     | atributos de los objetos de destino que deben cumplirse para que los permisos de acceso se aplican a los que accede el tema.  | ObjectAttributes | aggr | 0..1     |
| permiso                    | Los permisos asignados al objeto.   | Permiso          | attr | 0 .. *   |

|        |  |  |  |  |
|--------|--|--|--|--|
| Clase: | PermissionPerObject  |  |  |  |
|        | Los permisos son válidas para todos los sujetos como se especifica en la regla de permiso de acceso. |  |  |  |

| Clase:                     | ObjectAttributes  |             |          |          |
|----------------------------|---|-------------|----------|----------|
| Explicación:               | Un conjunto de elementos de datos que describen atributos de los objetos. Estos atributos deben hacer referencia a un elemento de datos dentro de una sub-modelo existente. |             |          |          |
| Hereda de:                 | --  |             |          |          |
| Atributo (* = obligatorio) | Explicación   | Tipo        | Tipo     | Tarjeta. |
| objectAttribute *          | A los elementos de datos que clasifica, además, un objeto.  | DataElement | árbitro* | 1 .. *   |

| Clase:                     | Permiso  |                     |          |          |
|----------------------------|--|---------------------|----------|----------|
| Explicación:               | Descripción de un solo permiso.  |                     |          |          |
| Hereda de:                 | --   |                     |          |          |
| Atributo (* = obligatorio) | Explicación  | Tipo                | Tipo     | Tarjeta. |
| permiso*                   | Referencia a una propiedad que define la semántica del permiso.<br><br><u>Restricción aasd-010:</u> La propiedad tiene la categoría de "constante".<br><br><u>Restricción aasd-011:</u> La propiedad autorización deberá ser parte del submodelo que se hace referencia en el "selectablePermissions" atributo de "AccessControl". | Propiedad           | árbitro* | 1        |
| kindOfPermission *         | Descripción del tipo de permiso. Posible tipo de autorización también incluye la negación del permiso. Valores:<br><br><ul style="list-style-type: none"> <li>• permitir</li> <li>• negar</li> <li>• no aplica</li> <li>• indefinido</li> </ul>  | PermissionKind attr |          | 1        |

| Clase:                     | SubjectAttributes   |             |          |          |
|----------------------------|---|-------------|----------|----------|
| Explicación:               | Un conjunto de elementos de datos que clasifica además un tema específico.  |             |          |          |
| Hereda de:                 | --  |             |          |          |
| Atributo (* = obligatorio) | Explicación   | Tipo        | Tipo     | Tarjeta. |
| subjectAttribute *         | Un elemento de datos que clasifica además un tema específico. Restricción aasd-025: El elemento de datos será parte del submodelo que se hace referencia en el "selectableSubjectAttributes" atributo de "AccessControl". | DataElement | árbitro* | 1 .. *   |

|              |  |
|--------------|--|
| Enumeración: | PermissionKind   |
| Explicación: | La enumeración de la clase de permisos que se le da a la asignación de un permiso a un sujeto. |
| Literal      | Explicación  |
| permitir     | Permitir que el permiso dado al tema.  |
| negar        | Explícitamente negar el permiso dado al tema.  |
| no aplica    | La autorización no es aplicable a la materia.  |
| indefinido   | Es sin definir si se permite el permiso, no es aplicable o negado al sujeto.                   |

## 6 Paquete Formato de archivo para el Activo Shell de administración (AASX)

## 6.1 General

En algunos casos de uso es necesario para el intercambio de la estructura total o parcial de la Concha de administración de activos con o sin valores asociados y / o hacer que la información persistente (por ejemplo, almacenarlo en un servidor de archivos). Esto significa que es necesario definir un formato de archivo que puede contener y almacenar esta información. Por lo tanto, se define un formato de archivo de paquete a la Concha de Administración de Activos (AASX) en base a los siguientes requisitos:

- Formato de archivo de paquete genérico para incluir la estructura de Shell de administración de activos, datos y otros archivos relacionados
- casos de uso principales son el intercambio entre organizaciones / socios y almacenamiento / persistencia de la información de las cáscaras de la Administración de Activos.
- Sin ninguna restricción legal y sin regalías. Preferentemente basada en una norma internacional con altas garantías de futuro mantenimiento de ese formato
- Existencia de APIs para crear, leer y escribir este formato
- firmas y las capacidades de cifrado digital se debe proporcionar
- Las políticas para la autenticidad y la integración de los archivos del paquete <sup>14</sup>

## 6.2 La selección del formato de referencia para el Shell de administración de activos formato de paquete

los ZVEI Führungskreis Industrie 4.0 - Spiegelgremium Modelle y Estándares ha decidido utilizar las Convenciones de empaquetado abierto (OPC) <sup>15</sup> formato que el formato de referencia para la definición del paquete de Shell Administración de Activos, debido a las siguientes razones:

- Convenciones de empaquetado abierto es una norma internacional específica en la norma ISO / IEC 29500-2: 2012 y ECMA-376.
- Convenciones de empaquetado abierto se basa en la postal (como un contenedor de paquete) y XML (para la descripción de algunos archivos internos y definiciones). Estas dos tecnologías son los más ampliamente utilizados en sus respectivos dominios y también se abordan para el archivado a largo plazo.
- Convenciones de empaquetado abierto se pueden utilizar como paquete de aplicaciones que no son de oficina también (hay muchos ejemplos disponibles, tales como paquetes NuGet, IED, etc.). Proporciona un modelo lógico que es independiente de cómo los archivos se almacenan en el paquete. Este modelo lógico se puede ampliar a cualquier tipo de aplicación.
- Convenciones de empaquetado abierto también se utiliza en el ámbito de la industria (por ejemplo, paquetes de IED, MTP - Namur Modul Tipo de paquete) y actualmente en discusión posible formato contenedor para algunos casos de uso FDT® y ODVA Proyecto XDS <sup>TM</sup>.
- Convenciones de empaquetado abierto (y paquetes Open Document Format) también es compatible con la firma digital. Se puede hacer para archivos individuales dentro del paquete. El cifrado no se especifica en los convenios de empaquetado abierto (sólo menciona lo que no se hace). De todos modos, el cifrado es todavía posible (ver más adelante)
- Hay algunas API para manejar paquetes de Convenciones de empaquetado abierto (API de Windows, .NET, Java, ...) sin necesidad de mucho conocimiento sobre la especificación técnica
- Fragmentación en los convenios de empaquetado abierto se anima, es decir, dividir archivos en trozos pequeños. Esto es mejor para reducir el efecto de la corrupción de archivos y un mejor acceso a los datos.
- Hay algunas organizaciones internacionales que recomiendan el uso de Open Document Format (ISO / IEC 26300-3) en lugar (por ejemplo, la UE, la OTAN, ...), pero esta recomendación está relacionada con los formatos utilizados específicamente en aplicaciones de oficina.
- El Office Open XML y especificaciones Convenciones de empaquetado abierto originaron de Microsoft Corporation y más tarde normalizados en la norma ISO / IEC 29500 y ECMA-376. Las versiones actuales y futuras de la norma ISO / IEC 29500 y ECMA-376 están cubiertos por la promesa de especificaciones abiertas de Microsoft, por lo que Microsoft "irrevocablemente promete"

<sup>14</sup> políticas basadas en roles para acceder a este paquete no está definido, ya que esta es una característica de los sistemas que alojan el EAA (véase la sección 0)

<sup>15</sup> No debe confundirse con OPC (Open Plataforma de Comunicaciones) de la Fundación OPC. Por lo tanto, vamos a utilizar el término completo de "Convenciones de empaquetado abierto" en lugar de la abreviatura "OPC".

no hacer valer cualquier reclamación contra los que hacer, usar y vender conformes implementaciones de cualquier especificación cubierto por la promesa (siempre y cuando aquellos que aceptan el estribillo promesa de demandar Microsoft por violación de patentes en relación con la implementación de Microsoft de la especificación cubierta). [24]

- Office Open XML (incluyendo el formato de Convenciones de empaquetado abierto) y el Formato de Documento Abierto formatos son políticamente conflictivos (ver detalles en [25] y [26]). La elección de Convenciones de empaquetado abierto como la opción para almacenar la información de Shell de administración de activos era exclusivamente una decisión técnica sobre la base de los argumentos mencionados aquí.
- Convenciones de empaquetado abierto se eligió a favor de iIRDS (v1.0). El alcance de iIRDS podría no estar alineado con los requisitos de la Administración de activos de Shell, es decir iIRDS es sobre todo un formato de almacenamiento de la documentación técnica de los dispositivos de la industria basadas en conceptos de la ontología.

### 6.3 Conceptos básicos de las Convenciones de empaquetado abierto

El modelo de embalaje especificada por las Convenciones de empaquetado abierto describe paquetes, partes, y relaciones.

Paquetes de mantenimiento de piezas, que contienen contenidos y recursos, tales como archivos <sup>dieciséis</sup>. Cada archivo en un paquete tiene un nombre de archivo

URIcompliant único, junto con un tipo de contenido especificado expresado en la forma de un tipo de medio MIME. Las relaciones se definen para conectar el paquete de archivos, y para conectar varios archivos en el paquete. La definición de las relaciones es la modelo lógico del paquete. El recurso que es una fuente de una relación debe ser el propio envase o un componente de datos (archivo) en el interior del paquete. El recurso de destino de una relación puede ser cualquier recurso URIaddressable dentro o fuera del paquete. Es posible tener más de una relación que comparten el mismo archivo de destino (véase el ejemplo 9-6 en la norma ISO / IEC 29500-2: 2012). los Modelo físico mapas de los conceptos lógicos a un formato físico. El resultado de esta asignación es un formato de paquete físico (un formato de archivo ZIP) en la que aparecen los archivos en una jerarquía de directorios similar. Cualquier persona u organización puede diseñar un formato de paquete físico mediante la asignación de los conceptos de paquetes lógicos para un formato físico deseado. Por lo tanto, los diseñadores de formato de paquete pueden diseñar y optimizar un formato físico para las necesidades específicas de una aplicación, sin comprometer la estructura lógica del paquete (adaptado de [27] y [28]).

### 6.4 Convenciones para el formato de archivo del paquete de Shell Administración de Activos (AASX)

El formato de Activos Administración Shell paquete (AASX) se deriva de las normas Convenciones abrir el envase, en consecuencia, que hereda sus características. Sin embargo, algunas convenciones se define para el AASX: \_\_\_\_\_

- formato de paquete y las reglas de acuerdo con la norma ISO / IEC 29500-2: 2012. Cualquier formato derivado de este estándar (tal como el formato de AASX) requiere la definición de un modelo lógico, modelo físico y un modelo de seguridad. Dichos convenios específicos se describen en las siguientes subsecciones.
- extensión del formato de AASX presentar: `.aasx`
- Tipo MIME para el formato AASX: `application / activo-administración-shell-paquete` <sup>17</sup>
- Icono para la AASX (a definir).
- El formato AASX puede ser identificado por la extensión del archivo y el tipo MIME. En cuanto al contenido, es posible identificar que al leer los archivos `/_rels/.rels` primera relación (como se define en los Convenios de empaquetado abierto) y en busca de un tipo de relación `http://www.admin-shell.io/aasx/relationships/aasx-` ( que es el punto de entrada para el modelo lógico de la Shell Administración de activos).
- Las siguientes rutas de acceso y nombres de archivo en el paquete ya están reservadas por la especificación de Convenciones de empaquetado abierto y por lo tanto no se utilizarán para cualquier formato derivado: `/[Content_Types].xml`

<sup>dieciséis</sup> El término "archivo" se utiliza en lugar de "parte".

<sup>17</sup> El tipo MIME currently es provisoria y necesita ser solicitado oficialmente.



`/_rels/.rels / < ruta_archivo> / _REL / < nombre de archivo>. REL (donde < nombre de archivo> es un archivo (incluyendo su extensión) en el paquete que es la fuente de las relaciones y`

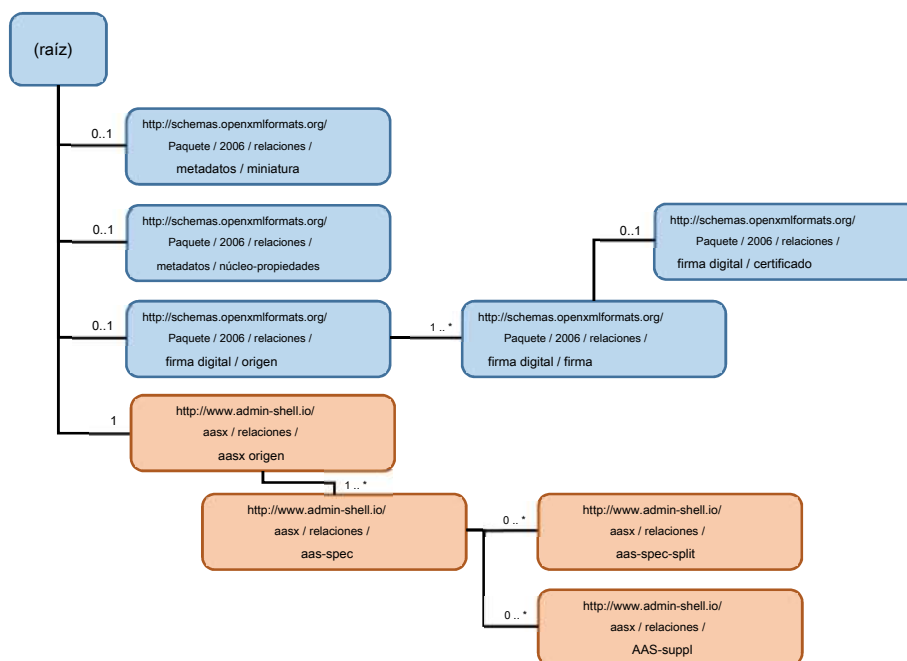
`<Ruta_archivo>` es la ruta de acceso a ese archivo.

- No es obligatorio para abrir el formato AASX en cualquier Office Open XML / abrir Convenciones de empaquetado compatibles oficina-aplicación existente (por ejemplo Microsoft Office, LibreOffice), debido a que las relaciones y los archivos necesarios para los diferentes "modelos" de oficina pueden no estar presentes (por ejemplo, <http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument> para el documento "docx").

## 6.5 modelo lógico

Como se mencionó antes, es necesario definir un modelo lógico para los formatos en la parte superior de los convenios de empaquetado abierto. Figura 53 define el modelo lógico para el formato AASX. Está hecho de un conjunto de tipos de relaciones (URI), su cardinalidad (¿cómo son posibles muchas relaciones de ese tipo) y la fuente de la relación. Además (no mostrado en la Figura 53), una instancia de relación específica tiene también un identificador único y un recurso de destino (URI de un archivo de destino dentro o fuera del paquete).

Figura 53 modelo lógico para el formato AASX <sup>18</sup>



Las relaciones de miniatura, de núcleo de propiedades, las firmas digitales (origen, firma y el certificado) están definidos en los Convenios de empaquetado abierto, por lo que no hay necesidad de reinventar. Las otras relaciones fueron definidos específicamente para apoyar los archivos específicos de Shell de administración de activos. A continuación una breve descripción de cada relación <sup>19</sup> de la figura 53:

- miniatura** - Opcional. Requeridos para definir una imagen en miniatura para ese paquete (por ejemplo, imagen del dispositivo administrado). La imagen en miniatura se puede mostrar en lugar del icono del paquete basado en la extensión y / o tipo de MIME.
- de núcleo de propiedades** - Opcional. Hay un esquema para describir el paquete a través de "propiedades del núcleo", que utiliza seleccionan elementos de metadatos Dublin Core, además de algunos elementos de empaquetado abierto Convenios-específicos. Las principales propiedades-no describen el Shell de administración, pero el propio envase. Algunos elementos de los coreproperties pueden ser similares / igual a elementos de la Concha de Administración. Algunos de núcleo de propiedades son: título, tema,

<sup>18</sup> Tenga en cuenta que el modelo lógico no dice nada sobre el formato / contenido de los archivos de destino de las relaciones. Este problema se solucionará en el modelo físico.

<sup>19</sup> Para evitar los largos nombres de las relaciones, vamos a utilizar el nombre corto a lo largo del texto.

Creador, palabras clave, descripción, lastModifiedBy, Revisión, LastPrinted, crear, modificar, Categoría, Identificador, ContentType, idioma, versión ContentStatus.

- **firma digital / origen, firma digital / firma y firma digital / certificado** - Opcional. Se requiere si usted necesita para firmar archivos y las relaciones dentro del paquete. Sus relaciones se dirigen básicamente archivos que contienen los datos de firmas (por ejemplo, certificado, digiere, ...). Ver la descripción más adelante en este documento acerca de las firmas digitales.
- **aasx origen** - Obligatoria. Origen de las relaciones y los archivos específicos AASX. A partir de este origen de uno o más miembros de AA pueden ser definidos. El productor no debe crear ningún contenido en el archivo de origen aasx sí. Como el Abierto de Convenciones de empaquetado relación modelo no permite apuntar directorios dentro del código postal, la alternativa es crear un archivo vacío que sirve como punto de entrada para la información AASX (este es el mismo enfoque que se utiliza para las firmas digitales) .
- **aas-spec** - al menos una relación de este tipo es obligatorio. Metas del archivo que contiene la estructura / especificación de una AAS (como se define en este documento) y por lo que sirve como punto de entrada de los datos específicos de AAS. Opcionalmente, algunas de la especificación de un AAS puede "estar de todas formas" en archivos separados, pero en cualquier caso, esta AAS-especificación de archivo sigue siendo obligatorio y contiene, al menos, la información no separable.
- **aas-spec-split** - Opcional. Esta relación se dirigirá a un archivo que contiene una parte separable de la especificación de AAS. Algunos formatos de serialización permiten que las partes del AAS puede ser dividido en varios archivos. Esos archivos son referenciados por este tipo de relación, de modo que cualquier consumidor de la AASX puede "volver a montar" la información AAS.
- **AAS-suppl** - Opcional. Dianas cualquier archivo adicional, sobre todo si se hace referencia (no almacena como blob) en los datos de un AAS (a través de la propiedad del archivo).

Nota: no todas las referencias dentro de la especificación de un AAS puede apuntar también archivos que se almacenan dentro de la AASX.

## 6.6 Modelo físico

El modelo físico define cómo se almacenan los diferentes archivos en el paquete, sobre la base de Convenciones de empaquetado abierto y cómo los archivos se abordan en las relaciones. Como se mencionó antes, el formato de paquete físico es un archivo ZIP que se puede abrir y editar en cualquier / postal aplicación compatible PKWARE. A fin de utilizar los identificadores de Shell y submodelos Administración, nombres descriptivos son requeridos. El nombre descriptivo de dichas entidades se construye mediante la búsqueda en todos los caracteres del identificador, que no son letras o dígitos y sustituyéndolas con un guión bajo "\_".

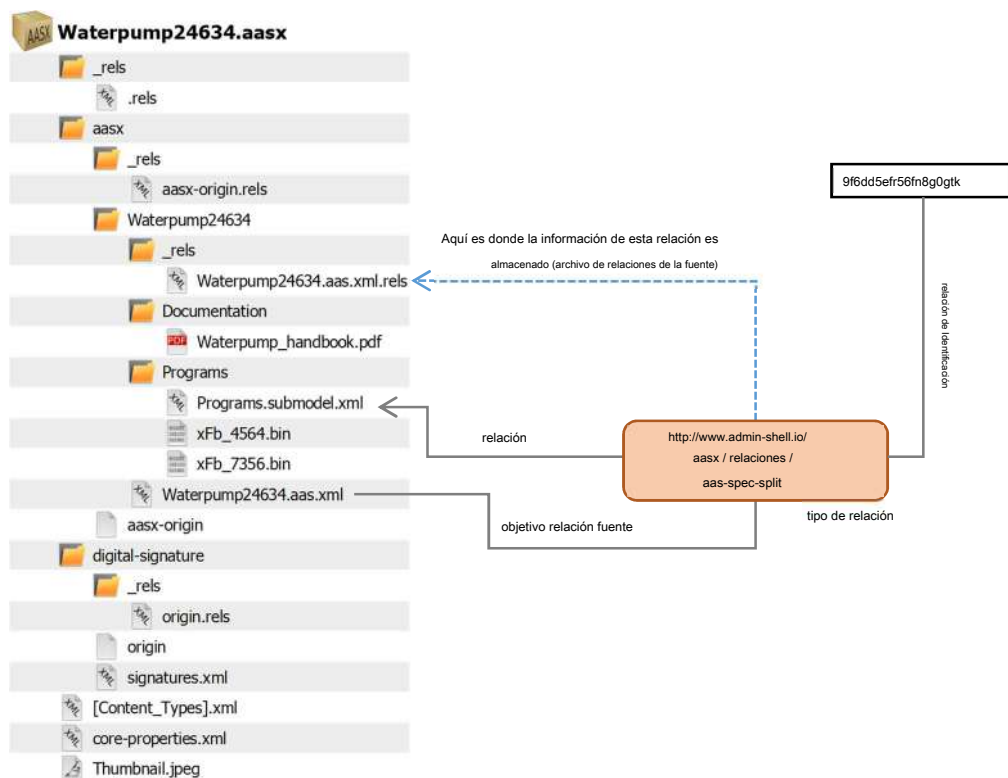
Una característica de este modelo físico y lógico es que el nombre de archivo y la ubicación de estos archivos puede ser personalizado (si relaciones asociadas tienen un URI correcto a esos archivos, y por lo tanto se pueden utilizar para localizar los archivos de acuerdo con la estructura lógica). Por ejemplo, un productor paquete puede almacenar un archivo aas-spec en /aasx/device.xml, el otro en /asset-admin-shell/productX123.xml, pero ambos utilizan el mismo tipo de relación de que los archivos. Para tener un enfoque más coherente en el modelo físico, los siguientes mejores prácticas está definido para el almacenamiento de archivos dentro del paquete AASX:

- 
- Convenciones de empaquetado abierto los archivos relacionados se deben almacenar de acuerdo con la API que se utilizó para generar / manipular el paquete AASX (no se recomienda hacer esto manualmente).
  - **/ Aasx** / será la carpeta raíz de la información específica del paquete AASX.
  - **/ Aasx / aasx-origen** será el destino de la relación aasx origen sin contenido (archivo vacío).
  - **/ Aasx / < aas-friendly-name>** / será la carpeta para almacenar todos los archivos para un AAS específica, donde **< aas-friendlyname>** es el nombre descriptivo de la AAS.
  - **/ Aasx / < aas-friendly-name> / <aas-friendly-name>. aas. <extensión>** será el objetivo de una relación de aas-spec tipo, donde **< extensión>** es la extensión en función del tipo de serialización (por ejemplo .xml, .json).
  - **/ Aasx / < aas-friendly-name> / <aas-friendly-name>. <vista-idshort>. vista. <extensión>** será el objetivo de una relación de tipo aas-spec-split que contiene una vista de las definiciones específicas de un AAS. Esto sólo es necesario si las definiciones de vista no están definidos en el archivo de destino de AAS-spec.

- **/ Aasx / < aas-friendly-name> / <aas-friendly-name>. CDIC. <extensión>** deberá ser el objetivo de una relación de tipo **aas-spec-split** que contiene la definición de un ConceptDictionary AAS. Esto sólo es necesario si el ConceptDictionary no está definido en el archivo de destino de AAS-spec.
- **/ Aasx / < aas-friendly-name> / <aas-friendly-name>. secattrib. <extensión>** deberá ser el objetivo de una relación de tipo **aas-spec-split** que contiene el modelo de seguridad de un AAS. Esto sólo es necesario si los modelos de seguridad no está definido en el archivo de destino de AAS-spec.
- **/ Aasx / < aas-friendly-name> / <submodelo-friendly-name> /** será la carpeta para almacenar los archivos relacionados con un submodelo de un AAS (objetivos de relaciones **AAS-suppl** que se hace referencia en ese submodelo y divisiones que contienen información submodelo). **< submodelo-friendly-name>** es el nombre descriptivo del submodelo.
- **/ Aasx / < aas-friendly-name> / <submodelo-friendly-name> / <submodelo-friendly-name>. submodelo. < extensión>** deberá ser el objetivo de una relación de tipo **aas-spec-split** que contiene una definición submodelo de un AAS. Esto sólo es necesario si un sub-modelo no está definido en el archivo de destino de AAS-spec.
- Esos nombres de archivo sólo debe contener caracteres que se pueden utilizar para nombres de archivo.
- Las convenciones definidas aquí no serán utilizados para otros archivos. Por ejemplo, cualquier otro archivo en una carpeta submodelo no contendrá el ".submodel." Subcadena en su nombre.
- Tenga en cuenta que el formato de los archivos seleccionados por el relaciones **aas-spec** y **aas-spec-split** dependen del formato de serialización que se utilizó para generar ellos (por ejemplo, XML, JSON, ...).
- También es posible tener diferentes formatos de serialización de la misma Administración Shell almacenado en el mismo AASX. En este caso, los diferentes formatos de serialización se pueden almacenar en paralelo con diferente extensión, diferentes tipos MIME y diferentes relaciones. Por ejemplo, **Waterpump24634.aas.xml** y **Waterpump24634.aas.json** se almacenan en la misma carpeta **Waterpump24634**, pero son objetivos de diferentes relaciones (diferentes ID) de la misma **aas-spec** tipo de relación. Ambos son entonces el punto de entrada de las diferentes ramas de relación fuente (cada uno con su propio archivo .rels, es decir **Waterpump24634.aas.xml.rels** y **Waterpump24634.aas.json.rels**).
- Para evitar la duplicación de datos, es posible orientar el mismo archivo por diferentes relaciones de (por ejemplo, dos relaciones diferentes de tipo **AAS-suppl** pueden tener el mismo archivo de destino).

Un ejemplo de un modelo físico para un AASX basado en un producto de muestra se muestra en la Figura 54. Se muestra el contenido del paquete que aparece en una vista de árbol y una cartografía ejemplo para el modelo lógico tal como se define en la Figura 53. La estructura física es basado en la mejor práctica se ha mencionado antes. Tenga en cuenta que en el ejemplo sólo hay un AAS en el paquete, un submodelo (programas) se almacena en un archivo separado y el certificado se incrusta en el archivo de firma (por lo que no hay necesidad de la relación adicional). También se supone en este ejemplo que los archivos de especificación de AAS son serializados en XML.

Figura 54 modelo físico para un AASX basado en un producto de muestra (izquierda) y un ejemplo de mapeo para el modelo lógico (derecha)



Es posible clasificar los archivos en un paquete AASX en los siguientes tipos: 1) los archivos que se hace referencia en las relaciones del modelo lógico y debe coincidir con el URI de destino dentro de cada relación, 2) "Los archivos que no son de origen o destino de cualquier relación (no permitido, ya que no siguen el modelo lógico definido en este documento y podrían afectar a algunos aspectos relativos a la firma digital y su verificación) y 3) Convenciones de empaquetado abierto archivos específicos que no están asociados con el modelo lógico (relaciones) :

- **/[Content\_Types].xml** - contiene una lista de extensiones y tipo MIME de todos los tipos de archivos dentro del paquete. La anulación elemento puede especificar el tipo MIME para archivos específicos independiente de la extensión. El tipo MIME de los archivos aasd específicos depende del tipo de serialización que se utilizó para generar el contenido de los archivos (por ejemplo, si se utiliza XML para algunos archivos, a continuación, el tipo MIME "text / xml" junto con las extensiones de archivo deberá aparecer en [Content\_Types] .xml. Si no hay ningún tipo MIME específico para algunos archivos, a continuación, "application / octet-stream" se utilizará.
- **/\_rels/.rels** - contiene todas las relaciones procedente de la fuente "raíz" (que es el propio envase), la fuente de la unión con una diana (el URI de un archivo interno o externo de recursos). Por ejemplo, para la relación miniatura que tiene este aspecto:
- **<Relación type = "http://schemas.openxmlformats.org/package/2006/ relaciones / metadata / miniatura" Target = "/Thumbnail.jpeg" id = "Rc76d59d18bd7440f" />**
- Esto significa que los datos de destino para esta relación miniatura se almacenan en /Thumbnail.jpeg.
- **/ < ruta\_archivo> / \_REL / < nombre de archivo>**. REL - las relaciones no root se almacenan en los archivos. Por ejemplo, la relación basada en el tipo de AAS-spec-split a partir del archivo fuente Waterpump24634.aas.xml se almacenan en archivos /aasx/Waterpump24634/\_rels/Waterpump24634.aas.xml.rels.

Un AASX se puede generar mediante el uso de diferentes medios:

- Manualmente mediante la adición de archivos (cambio de archivos) a (de) un archivo Zip. Esto requiere un profundo conocimiento del formato de Convenciones de empaquetado abierto, debido a la adición de sólo un archivo en el paquete con un editor postal no es suficiente (es decir, que tenga que editar los .xml [Content\_Types] y algunos de los archivos .rels también)

- Programación generar y cambiar el formato de paquete (por ejemplo, usando System.IO.Packaging .NET). Normalmente, esto evitará los errores que se pueden hacer al crear manualmente el paquete. Además, se deben considerar los procedimientos específicos abiertos Convenciones de empaquetado, el modelo lógico, física y la seguridad definidos para el AASX derivada.

## 6.7 Firmas digitales

Esencialmente, la firma digital de un documento electrónico (en este caso, los archivos y las relaciones dentro del contenedor) tiene como objetivo cumplir los requisitos siguientes [29]:

- que el destinatario pueda verificar la identidad del remitente (autenticidad);
- que el emisor no puede negar que firmó un documento (no repudio);
- que el destinatario es incapaz de inventar o modificar un documento firmado por otra persona (integridad).

Una firma digital no "bloquear" un documento o hacer que se encriptado (aunque ya se puede cifrar). el contenido del documento no ha cambiado después de haber sido firmado. Las firmas digitales no impiden contenido firmado de ser vistos por los consumidores no deseados.

Una característica la firma digital ya está prevista por la especificación Convenios abierto de embalaje [27]. Este marco de firma de paquetes utiliza el XML Digital Signature Standard, tal como se define en la Recomendación W3C Firma XML Sintaxis y Procesamiento. Esta recomendación especifica la sintaxis XML y las reglas de procesamiento para producir y almacenar las firmas digitales.

- Los archivos de paquetes definidos para el marco de la firma son el archivo de origen, el archivo (s) firma, y el archivo (s) certificado.
- **firma digital de archivos / origen - punto de partida para navegar a través de las firmas en un paquete.** El archivo de origen está dirigido desde la raíz de paquete utilizando la relación origen firma digital (como se muestra en el modelo lógico en la Figura 53). múltiples archivos de firma se pueden dirigir desde el archivo de origen. Si no hay firmas en el paquete, el archivo de origen no estará presente.
- **firma digital / archivo de firma (s) - contener texto definido en el estándar W3C Firma Digital, así como en el espacio de nombres de envasado.** Los archivos están dirigidos desde el archivo de origen con la relación de firma (como se muestra en el modelo lógico en la Figura 53).
- **firma digital archivo de certificado / (s) - El certificado X.509 requerida para identificar al firmante, si se coloca en el paquete, puede ser embebido dentro de un** archivo de firma, o se almacena en un archivo de certificado separado. El archivo de certificado opcional está dirigido desde un archivo de firma con la relación de certificados. El archivo de certificado se puede compartir entre múltiples archivos de firma.

En el paquete, los archivos y las relaciones individuales pueden ser firmados de forma independiente <sup>20</sup>, es decir que es posible seleccionar qué archivos y las relaciones necesitan una firma y cuyo certificado para ser utilizado para firmar. Cuando el archivo de relaciones (.rels) se firmó en su conjunto, todas las relaciones definidas en ese archivo se firman también. Además, es posible utilizar más de un certificado para firmar archivos y las relaciones.

El marco firma convenios de embalaje abierta es bastante flexible, y por lo tanto debe tomarse algunas consideraciones, especialmente cuando la definición de políticas. La especificación de Convenciones de empaquetado abierto no define políticas, sólo se menciona que "los diseñadores que incluyen firmas digitales deben definir las políticas de firma que son significativos para sus usuarios". Además de garantizar la autenticidad, no repudio e integridad, firmas digitales también se pueden utilizar para definir las **políticas que están destinados por los firmantes <sup>21</sup> (por lo general los productores de paquetes) o de acuerdo con los consumidores de paquetes (por ejemplo, los consumidores sólo aceptará paquete con contenido firmado). Las decisiones** tomadas durante el proceso de firma de impacto, que las operaciones consiguientes se pueden verificar (por ejemplo, permitiendo posterior a la modificación de un archivo, la adición de nuevas relaciones ...).

<sup>20</sup> Los archivos individuales y las relaciones pueden ser firmados, pero no el paquete completo. Se trata de una cuestión de definición, pero la firma del paquete completo podría significar para firmar todos los archivos dentro del paquete (excepto el archivo de firma).

<sup>21</sup> Las políticas que se describen aquí son para el paquete AASX y lo que se puede cambiar. No define ninguna política, por ejemplo, sobre el uso de un AAS.

No hay necesidad de un archivo separado en el paquete sobre las políticas, ya que esta información puede ser recuperada políticas sobre cómo se realiza la firma. La firma de un archivo específico en el paquete implícitamente expresar la intención del firmante de lo que es o no se permite que los archivos con archivos y afines (en el caso de los archivos de relación). Por ejemplo, la firma del archivo de relaciones aasx origen no va a permitir la adición de nuevos AAS al paquete. Si se añaden nuevos AAS todos modos, esto anularía la firma original y nadie puede culpar al firmante original para ese cambio. Un productor bulto deberá seguir una política de firma digital basada en las siguientes opciones:

1. **firmar nada**
2. **Firmar todo y siguiendo así la política de “No hay cambio permitido al paquete” de la Tabla 16.**
3. **Personal de acuerdo con la firma de una o más políticas de la Tabla 16.**

El consumidor paquete puede seguir un proceso de validación en base a la política del firmante (s) o una verificación interna del paquete de acuerdo a sus propias políticas. La política de firma definido por el firmante (s) no dice directamente que el consumidor debe validar el paquete, pero dice cómo se pretende por el firmante (s). Sin embargo, la validación podría ser obligatorio para las aplicaciones conjuntas en varias partes (productores y consumidores de paquetes) tienen que seguir las mismas reglas. El siguiente proceso para la validación<sup>22</sup> AASX paquetes se establece:

1. El proceso de validación debe comenzar por comprobar que el paquete se consume es de acuerdo con el Abierto de embalaje especificación convenios y que sus implementa el modelo lógico de acuerdo con la definición AASX. Opcionalmente se puede analizar si el modelo físico es de acuerdo con la mejor práctica para el almacenamiento de archivos dentro del paquete AASX.
2. Los archivos que no son de origen o destino de cualquier relación, no se les permite (además de las Convenciones de empaquetado abierto archivos específicos).
3. Después de estos pasos, los certificados existentes que se utilizaron para firmar el contenido del envase deberá válido y de confianza.
4. Todo el contenido firmado debe entonces ser verificada y válidos en contra de la información del certificado proporcionado.
5. El contenido firmado también revelará un conjunto de políticas<sup>23</sup> definido por los firmantes o definido de acuerdo con el varias partes (productores y consumidores del paquete), que deben ser seguidos por el consumidor cuando se cambia el paquete sin invalidar ella (véase la Tabla 16).
6. Un paquete es válido sólo si todos los pasos anteriores se realizan con éxito. Cualquier cambio hecho al paquete de el consumidor requiere una revalidación del paquete.

Cualquiera de los pasos mencionados en la validación se puede realizar de forma independiente sin los otros, pero al hacerlo, no se considera como la validación (por ejemplo, proceso de verificación interna por un consumidor sólo puede requerir para comprobar si el paquete está de acuerdo con los convenios de empaquetado abierto e implementación del modelo lógico AASX, sin verificar las firmas).

Tabla 16 Conjunto de políticas posibles en función de cómo se firman archivos de paquetes, cómo habilitar una determinada política y las consecuencias de una política

| Política | Cómo habilitar la directiva        | Consecuencia   |
|----------|------------------------------------|--|
| General  | Ningún cambio permitido al paquete | Firmar todos los archivos y las relaciones en el paquete (con excepción de .xml [Content_Types] <sup>24</sup> y el archivo de firma (s))   |
|          |                                    | Invalida cualquier cambio en los archivos firmados. Los nuevos archivos que se agregan posteriormente no tienen una firma y no se mencionan en cualquier relación, lo cual invalida esos archivos. |

<sup>22</sup> Validación. La seguridad de que un producto, servicio o sistema cumple con las necesidades de los clientes y otras partes interesadas identificadas. A menudo implica la aceptación y adecuación con los clientes externos. Contraste con la verificación, que es a menudo un proceso interno. (Adaptado de La guía PMBOK, una norma adoptada por IEEE, 4ª edición)

<sup>23</sup> Estas políticas son para el paquete AASX y no por el propio AAS.

<sup>24</sup> Cuando se lee un paquete AASX, no se basan en la TruStability del archivo [Content\_Types] .xml, ya que no era posible firmar este archivo.

|                 |   |   |   |
|-----------------|---|---|---|
|                 | Ningún cambio permitido el contenido del archivo X o eliminación del archivo X                          | Firmar el archivo X dentro del paquete (por ejemplo, AAS, un archivo submodelo, cualquier archivo, ...)   | Invalida la manipulación del contenido o supresión de archivo X.  |
|                 | Ningún cambio permitió una relación X   | Signo relación X  | Invalida manipulación o eliminación de la entrada relación (es decir, el tipo de relación, de la identificación y el objetivo URI) en el archivo de relación correspondiente. Esto no invalidará el contenido de los archivos de origen y de destino de una relación, una vez manipulado.   |
|                 | Ningún cambio permitido a cualquier relación que tienen archivo de origen X                             | Firmar archivo de relaciones X (X.rels)   | Invalida añadir, cambiar o quitar cualquier relación mencionada en dicho archivo de relaciones. Esto también se anulará la adición de nuevos archivos que son de otro modo objetivo en ese archivo de relación. Por ejemplo, si no existe una relación de la miniatura en el archivo de relaciones de raíz antes de la firma de ese archivo, a continuación, se invalida una adición posterior de relación miniatura. |
|                 | Permitir a las firmas digitales   | La relación de firma digital / origen debe ser firmado (alternativamente, firmar el archivo de relaciones completa de la raíz que contiene esta relación) | Permitirá a las firmas digitales (pero no especifica las reglas para la firma, por ejemplo si las nuevas firmas se pueden añadir).  |
|                 | Habilitar núcleo-propiedades  | La relación de metadatos / Core-propiedades debe ser firmado (alternativamente, firmar el archivo completa relación de raíz que contiene esta relación)   | Permitirá a los de núcleo de propiedades del paquete.   |
|                 | Activar miniatura   | La relación de metadatos / miniatura debe ser firmado (alternativamente, firmar el archivo completa relación de raíz que contiene esta relación)          | Permitirá a la miniatura para el paquete.   |
|                 | Prohibir contra-firmas (añadiendo nuevas firmas)  | Firmar el archivo de relaciones origen de firma   | Invalida contra-firmas.   |
|                 | Prohibir la modificación digiere archivo / relaciones existentes para la firma basada en un certificado | muestra objeto dentro del archivo de firma correspondiente que contiene resúmenes de todos los archivos / relaciones                                      | Invalida cualquier cambio en los compendios y adición de nuevos compendios de archivo.  |
| AASX específica | Habilitar especificación AASX   | El aasx origen debe ser firmado relación (alternativamente, firmar el archivo de relaciones completa de la raíz que contiene esta relación).              | Permitirá a la especificación AASX en la parte superior de las Convenciones de empaquetado abierto.   |
|                 | Prohibir la adición de un nuevo AAS o la eliminación de un AAS existente.                               | Firmar el archivo de relaciones aasx origen   | Invalida adición o eliminación de AAS.  |
|                 | Prohibir la adición de unas piezas nuevas divisibles o eliminación de uno ya existente a / desde un AAS | Firmar el archivo de relaciones de especificación aas   | Invalida la adición o eliminación de partes divisibles a / desde un AAS.  |
|                 | Prohibir la adición de un nuevo archivo complementario o eliminar una existente a / desde un AAS        | Firmar el archivo de relaciones de especificación aas   | Invalida la adición o eliminación de archivos adicionales a / desde un AAS.   |

## 6.8 encriptación

La especificación de Convenciones de empaquetado abierto (ISO / IEC 29500-2: 2012) menciona que "los paquetes basados en ZIP no incluirán encriptación como se describe en la especificación postal. ejecutores del paquete harán cumplir esta restricción. [M3.9]"<sup>25</sup>.

<sup>25</sup> La razón de esto podría estar relacionado con el requisito de transparencia para el formato de paquete, así como los requisitos de licencia de PKWARE. Para la norma ISO / IEC 21320-1 (contenedor de documentos del archivo: Core) existe la siguiente declaración: "El cifrado de archivos individuales y del directorio central está prohibido. De ahí que este perfil de ZIP\_PK es más transparente que el formato de los padres ". [30]

Sin embargo, un paquete de Convenciones de empaquetado abierto puede ser cifrada con otros medios y algunas aplicaciones que utilizan este formato de paquete como la base para un formato más específico, puede utilizar el cifrado durante el intercambio o DRM para su distribución. [24]

Un ejemplo es la estructura de documentos de Office Criptografía (MS-OFFCRYPTO) utilizado por los formatos de oficina de derivados. Algunas tecnologías utilizadas pueden ser cubiertos por las patentes de Microsoft y por lo tanto no se recomienda para el formato AASX. Gestión de derechos digitales (DRM) también puede ser utilizado para cifrar elementos de contenido en un paquete con los derechos de acceso específicos otorgados a autorizar a los usuarios (ver la aplicación en el espacio de nombres System.IO.Packaging [31]). En cuanto a la encriptación y la confidencialidad, se seguirán las siguientes reglas:

1. Decidir si existe la necesidad de incluir el contenido confidencial en un paquete. Si no hay ninguna razón, entonces el contenido confidencial no debe ser incluido.
2. Si se desea cifrado para un acto de comunicación temporal (por ejemplo, intercambio de correo electrónico, ...) o si un AASX necesita ser almacenado en algún lugar para que se pueda abrir más tarde por la misma entidad, a continuación, los métodos de cifrado pueden ser utilizados para ese medio específico (por ejemplo, uso BitLocker para el almacenamiento del AASX en sistemas basados en Windows que lo soportan, utilice S / MIME para el intercambio de correos electrónicos cifrados entre entidades, etc.).
3. Para todos los demás casos de uso<sup>28</sup> donde se requiere el cifrado para algunos o la totalidad del contenido de la AASX:
  - métodos de cifrado se pueden utilizar para los archivos individuales en el paquete AASX, tan pronto como la versión "cifrado" reemplaza el archivo original en el paquete, se conoce el tipo MIME del formato de cifrado, y el tipo MIME deben estar inscritos en el [contenido -Tipo] .xml. Las relaciones como se definen en este documento siguen siendo los mismos, siempre que el contenido está cifrado o no. Tenga en cuenta que se pueden abrir archivos relacionados Convenciones de empaquetado, así como archivos de relación no serán cifrados y firmas digitales deben llevarse a cabo después del cifrado. Un ejemplo de un estándar de cifrado es el MIME seguro (S / MIME), donde el contenido cifrado debe ser almacenado en formato de aplicación / pkcs7-mime como se define en RFC 5652 y utilizar la extensión de fichero \* .p7m.
  - Además de cifrar el contenido del paquete (archivos individuales) es posible cifrar el paquete completo (por ejemplo, utilizando también Secure MIME y guardar el paquete cifrado en formato de archivo de la aplicación / pkcs7-mime). En este caso, la firma del contenido del envase debe hacerse antes de la encriptación.

---

<sup>28</sup> Un caso de uso podría ser para cifrar un submodelo y sólo proporcionar el acceso a los datos sin cifrar después de pagar una cuota.





## 7 Resumen y Perspectivas

En este documento se define un metamodelo UML para el punto de vista estructural de la Concha AssetAdministration. Un esquema XML, así como para el esquema JSON se derivan de ella.

Adicionalmente, se proporciona una plantilla de especificación de datos para las propiedades que definen. Los siguientes aspectos

serán cubiertos por las próximas versiones del metamodelo estructural:

- Los depósitos específicos para la Administración de composite
- De registro de eventos e Historia (incluyendo aspectos de seguridad)
- aspectos de seguridad adicionales que aún no se necesitaban para el caso de uso en estudio como
  - o Autenticación
  - o certificados
- Las políticas de reajuste
- Plantillas adicionales de especificación de datos para tipos adicionales de elementos de submodelos
- La manipulación de las capacidades, habilidades y cómo se relacionan con la operación.
- Definición de un lenguaje formal fórmula de restricciones
- Más serializaciones como para AutomationML y RDF
- atributos adicionales para información administrativa, por ejemplo, "marca de tiempo", "tiempo creado" y "responsabilidad operativa" como se define en IEC 62832 etc.

Una serialización OPC UA se trabaja en el grupo de trabajo conjunto de la Fundación OPC, VDMA y ZVEI "I4AAS". El modelo y conceptos de la Concha de administración de activos descritos en esta publicación meta son, entre otros, implementados en el Kit de desarrollo de software de código abierto (SDK) del proyecto financiado con fondos públicos BaSys 4.0. Se compone de módulos para la creación, modificación y exportación (XML y JSON) de las Conchas de administración de activos, así como los demás. Que estará disponible a principios de 2019 <sup>27</sup>.

Las siguientes partes de la serie de documentos cubrirán:

- Interfaces y API para el uso de un único modelo de información AAS descrito en la Parte 1 (acceso, modificación, consulta y ejecutan información y funcionalidad activo)
- La infraestructura, que alberga múltiples interconexiones y AAS juntas. Implementa registro, servicios de descubrimiento, manejo de punto final y más.

---

<sup>27</sup> <https://projects.eclipse.org/projects/technology.basysx>

anexo

## Anexo A. Los conceptos de la Concha de Administración

### 1. General

En esta cláusula, la información general se da información sobre las fuentes de información y conceptos relevantes para el Shell de administración de activos. Algunos de estos conceptos se explican de una manera general. Algunos conceptos son la actualización con el fin de reflejar las decisiones de diseño reales. no se introducen nuevos conceptos. Por lo tanto, la cláusula puede ser tomado como un totalmente informativo (anexo) para la especificación de la Shell Administración.

### 2. fuentes y documentos relevantes

Los siguientes documentos fueron utilizados para identificar los requisitos y conceptos para el Shell de administración:

- Estrategia de ejecución de Plattform Industrie 4.0 [1,2]
- Aspectos del plan de investigación en escenarios de aplicación [7]
- Continuación de los escenarios de aplicación [8]
- Estructura de la Administración de Shell [4, 18]
- Ejemplos para el Shell administración de los componentes Industrie 4.0 [6]
- Descripción técnica general "Secure identidades" [9]
- Seguridad de la Concha de Administración [14]
- Las relaciones entre los componentes I4.0 - Componentes compuestos y la producción inteligente [12]

Nota: El glosario global Industrie 4.0 se puede encontrar en: <https://www.plattform-i40.de/I40/Navigation/EN/Service/Glossary/glossary.html>

Nota: La librería en línea de la Plataforma Industrie 4.0 se puede encontrar en: <https://www.plattform-i40.de/I40/Navigation/EN/InPractice/Online-Library/online-library.html>

### 3. Conceptos básicos para Industrie 4.0

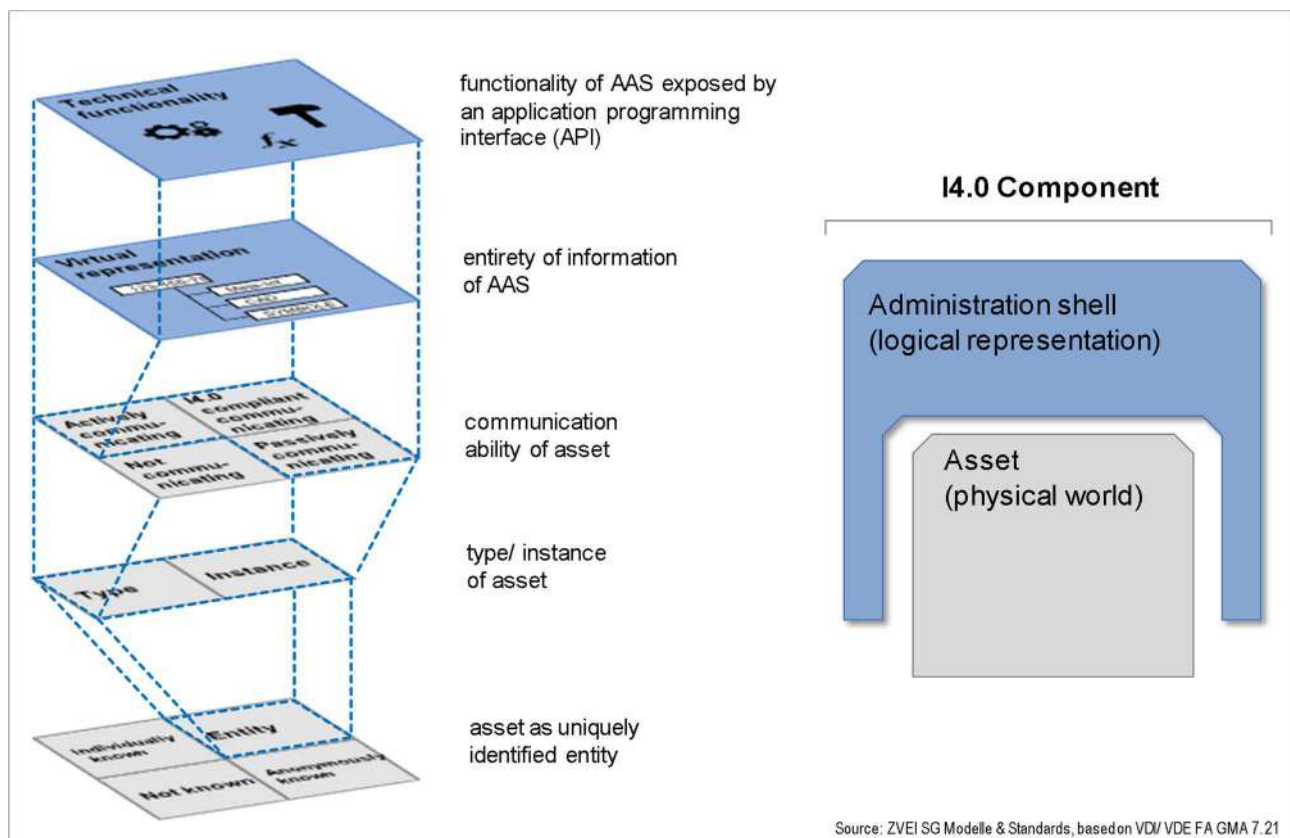
Industrie 4.0 se describen los conceptos y definiciones para el dominio de la fabricación inteligente. Para Industrie 4.0, el activo plazo, siendo cualquier "objeto que tiene un valor para una organización", es de importancia central [2, 23]. De este modo, los activos en Industrie 4.0 puede tomar casi cualquier forma, por ejemplo, un sistema de producción, un producto, una instalación de software, propiedad intelectual o incluso los recursos humanos.

Según [23], la "referencia arquitectura modelo Industria 4.0 (RAMI4.0) proporciona una vista estructurada de los elementos principales de un activo usando un modelo de nivel que consta de tres ejes [...]. Interrelaciones complejas pueden por lo tanto pueden desglosar en más pequeños, secciones más manejables mediante la combinación de los tres ejes en cada momento de la vida del activo para representar cada aspecto relevante ".

Activos tendrán una representación lógica en el "mundo de la información", por ejemplo, será administrada por sistemas informáticos. Por lo tanto, un activo tiene que ser identificado con precisión como una entidad, tendrá un "estado específico dentro de su vida (por lo menos un tipo o una instancia)", tendrán capacidades de comunicación, serán representados por medio de la información y deberá ser capaz de proporcionar funcionalidad técnica [23]. Esta representación lógica de un activo se llama Shell Administración [4]. La combinación de activos y Shell Administración constituye el llamado componente I4.0. En los documentos internacionales [18], el término de fabricación inteligente reemplaza el término Industrie 4.0.

Para la gran variedad de activos en Industrie 4.0, el Shell de administración permite la manipulación de estos activos en el mundo de la información de siempre de la misma manera. Esto reduce la complejidad y permite la escalabilidad. motivación adicional se puede encontrar en [2] [4] [7] [8].

Figura 55 conceptos importantes de Industrie 4,0 unidos al activo [2, 23]. I4.0 Componente a ser formado por Shell Administración y activo.



#### 4. El concepto de propiedades

Según [20], la "serie IEC 61360 proporciona un marco y un modelo de información para los diccionarios de productos. El concepto de tipo de producto está representado por 'clases' y las características del producto están representados por 'propiedades'". Tales propiedades son estandarizados elementos de datos. Las definiciones de estas propiedades se pueden encontrar en una gama de repositorios, tales como IEC CDD (diccionario de datos común) o eCI @ ss. La definición de una propiedad (aka datos estandarizados tipo de elemento, tipo de propiedad) asocia un identificador único a nivel mundial con una definición, que es un conjunto de atributos bien definidos. atributos relevantes para el Shell de administración son, entre otros, el nombre preferido, el símbolo, la unidad de medida y una definición textual legible de la propiedad.

Figura 56 definición ejemplar de una propiedad en el IEC CDD

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

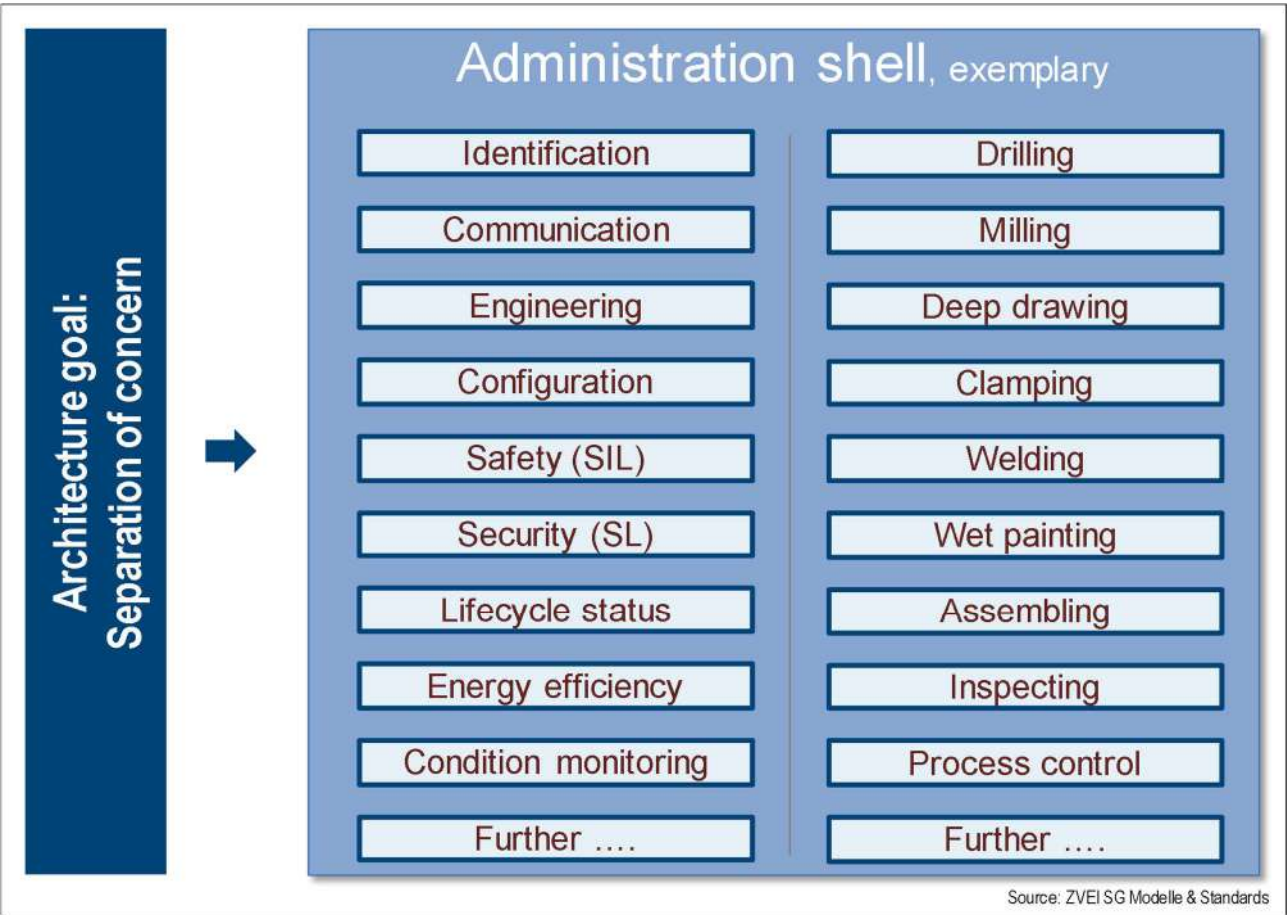
La creación de instancias de esta definición (justo 'propiedad', propiedad de instancia) normalmente asocia un valor a la propiedad. Por este mecanismo, la información semántica bien definida puede ser transmitida por la Administración de Shell.

**Nota:** Industrie 4.0 y fabricación inteligente, en general, se requieren muchas propiedades que están más allá del alcance actual de la norma IEC CDD, eCI @ ss u otros repositorios. Se espera, que serán introducidos estos conjuntos de propiedades, a medida que más y más dominios se modelan y (cláusula siguiente) estandarizado.

## 5. El concepto de submodelos

"La Administración de Shell es la representación digital estandarizada del activo, piedra angular de la interoperabilidad entre las aplicaciones de gestión de los sistemas de fabricación" [18]. Por lo tanto, se necesita para proporcionar una descripción mínima pero suficiente de acuerdo con los diferentes escenarios de aplicación en Industrie 4.0 [7] [8]. Muchas de las normas (internacionales), diferentes especificaciones del consorcio y las especificaciones del fabricante ya pueden contribuir a esta descripción [18]. Como muestra la figura, la información de diferentes diversos dominios técnicos podría estar asociado con un activo respectivo y por lo tanto, muchas propiedades diferentes se requieren para ser representado en las conchas de la administración de futuras Componentes I4.0. Con el fin de gestionar estas complejas conjunto de información, submodelos proporcionan una separación de preocupación.

Figura 57 Ejemplos de diferentes dominios de proporcionar propiedades de submodelos de la Shell Administración



La Shell Administración De este modo se compone de una serie de submodelos [4]. Estos representan diferentes aspectos del activo de que se trate; por ejemplo, pueden contener una descripción relativa a la seguridad o la seguridad [14], pero también podría esbozar diversas capacidades de proceso tales como la perforación o la instalación [6].

Desde el punto de vista de la interoperabilidad, el objetivo es estandarizar un solo sub-modelo para cada aspecto de dominio / técnico. Por ejemplo, este modo será posible encontrar una máquina de perforación mediante la búsqueda de un proyectil de administración que contiene un submodelo "perforación" con propiedades adecuadas. Para la comunicación entre diferentes componentes I4.0, ciertas propiedades a continuación, se puede suponer que existe. En un ejemplo como éste, un segundo submodelo, "eficiencia energética", entonces podría asegurar que la máquina de perforación es capaz de reducir su consumo de electricidad cuando no está en funcionamiento.

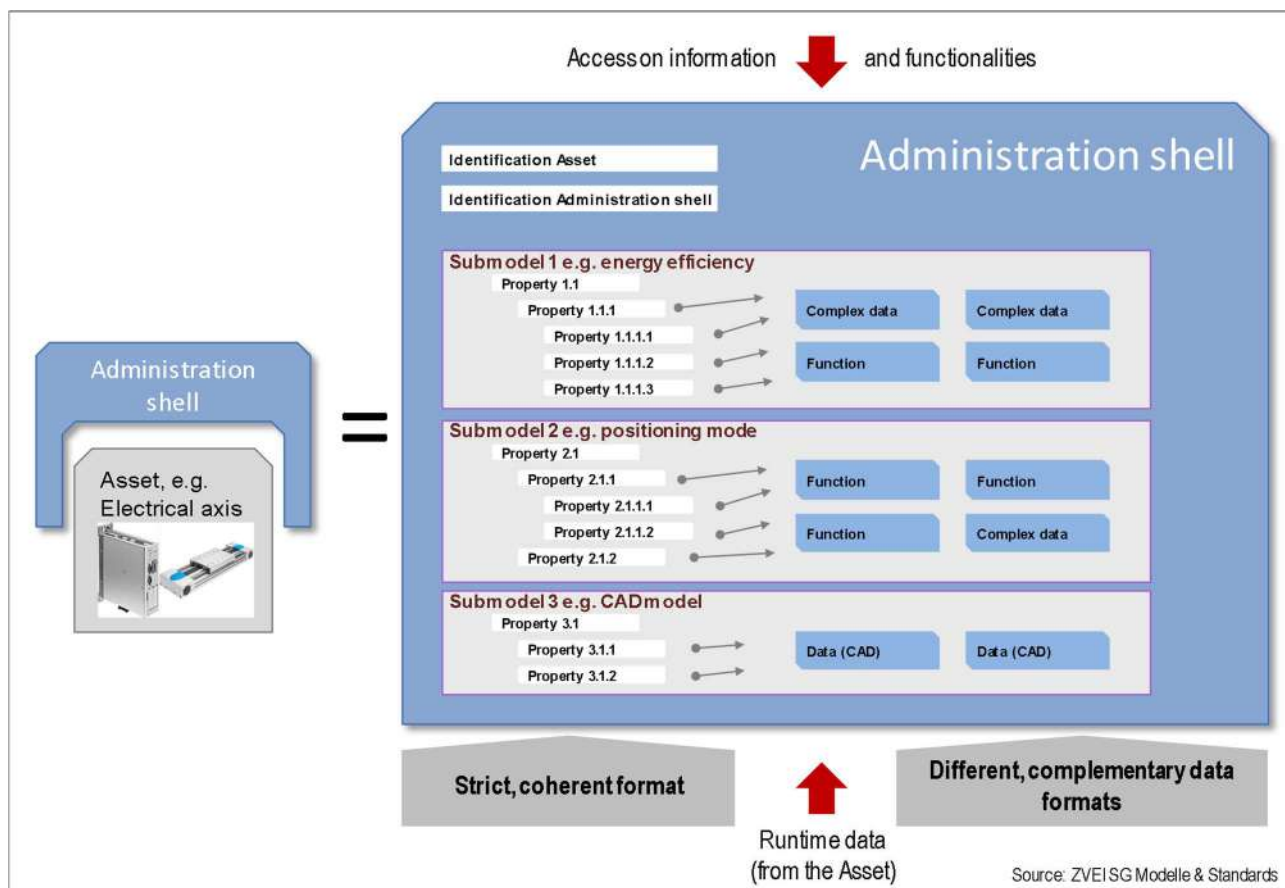
Nota: beneficio adicional de la Concha de Administración será simplificar la actualización de las propiedades de diseño de producto (y en particular herramientas de diseño del sistema), la actualización de las propiedades de los datos reales recogidos en los casos de los activos, mejorar la trazabilidad de los activos a lo largo del ciclo de vida y ayudar a certificar los activos de datos.

6. Estructura básica del Shell de administración de activos

El documento sobre la estructura de la Administración de activos de Shell [4] [18] presenta una visión aproximada, lógica de la estructura de la Shell AssetAdministration. La Shell AssetAdministration - se muestra en azul en la siguiente figura - comprende diferentes conjuntos de información. Ambos, el activo y el Shell de administración se identifican mediante un identificador único global. Comprende un número de submodelos para una caracterización de la Shell AssetAdministration.



Figura 58 Estructura básica de la Shell AssetAdministration



Propiedades, datos y funciones también contendrán información que no todos los socios dentro de una red de valor añadido o incluso dentro de una unidad de organización debe ser capaz de acceder o cuya integridad y disponibilidad debe ser garantizada. Por lo tanto, la estructura de la Concha de Administración deberá ser capaz de manejar aspectos como la protección de acceso, la visibilidad, la gestión de la identidad y los derechos, confidencialidad e integridad. seguridad de la información tiene que ser respetado y tiene que estar alineado con un concepto global de seguridad. Implementación de la seguridad debe ir de la mano con la aplicación de otros componentes de un sistema general.

Cada submodelo contiene una cantidad estructurado de propiedades que se refieren a los datos y funciones. Un formato estandarizado basado en la norma IEC 61360-1 / ISO 13584-42 está previsto en las propiedades. Por lo tanto, el valor de propiedad de definición deberá seguir los mismos principios que también la norma ISO 29002-10 y IEC 62832-2. Los datos y las funciones están disponibles en varios formatos, complementarios.

Las propiedades de todos los submodelos, por tanto, dan lugar a un directorio en constante lectura de la información clave de la Concha de Administración y por lo tanto del componente I4.0. Para habilitar la semántica de vinculación, conchas de administración, bienes, propiedades submodelos y todos deben ser claramente identificados. identificadores globales permitidas son IRDI (por ejemplo, en la norma ISO TS 29002-5, eCI @ ss e IEC Común diccionarios de datos) y URIs (Identificadores de Recurso Único, por ejemplo, para ontologías).

Debería ser posible filtrar los elementos de la Shell Administración o submodelos según diferentes puntos de vista dados ( •

Ejemplo C.4 en [18]). Esto facilita diferentes perspectivas o casos de uso para la aplicación de la información de la Administración de Shell.

## 7. requisitos

Esta sección recoge los requisitos de diversos documentos que tienen impacto en la estructura específica de la Concha de Administración. Estos requisitos sirven como entrada para la descripción específica de las estructuras de la Administración de Shell.

Los siguientes requisitos se toman del documento "Estrategia de ejecución de Plattform Industrie 4.0" [2]. Están marcados "COMIENZO". La columna "seguimiento" valida los requisitos mediante la vinculación a características del metamodelo UML o este documento en general.

| CARNÉ DE IDENTIDAD | Requisito   | Rastreo   |
|--------------------|---|---|
| COMIENZO # 1       | Una red de Industrie 4.0 componentes debe estar estructurado de tal manera que son posibles conexiones entre cualquier punto final (Industrie 4.0 componentes). Los componentes Industrie 4.0 y sus contenidos han de seguir un modelo semántico común.                     | RED posible, pero no alcance de esta parte de la serie de documentos. modelo semántico común <b>realizado por submodelos específicos de dominio ( HasSemantics / ConceptDescription y por Relaciones)</b> |
| COMIENZO # 2       | Debe ser posible definir el concepto de un componente Industrie 4.0 de tal manera que se pueda cumplir con los requisitos de diferentes esferas de actividad, es decir, "piso de la oficina" o "taller".  | En cuanto al contenido, muchos submodelos diferentes posible.   |
| COMIENZO # 3       | Industrie 4.0 de comunicación conforme debe realizarse de tal manera que los datos de una representación virtual de un componente Industrie 4.0 pueden mantenerse ya sea en el propio objeto o en un sistema IT (nivel superior).   | Metamodelo y la información de representación independiente de cualquier escenario de implementación.   |
| COMIENZO # 4       | En el caso de una representación virtual de un componente I4.0 en un sistema de nivel superior, una asociación integridad debe garantizarse entre el activo y su representación.  | parte integridad de enfoque de seguridad.   |
| COMIENZO # 5       | Un modelo de referencia adecuado debe establecerse para describir cómo un sistema de nivel superior se puede hacer que el Shell de administración disponibles en una forma compatible con 4.0 Industrie (enfoque SOA, el principio de delegación).                          | Alcance de la próxima parte de la serie de documentos; no alcance de esta parte.  |
| COMIENZO # 6       | Se requiere una descripción de cómo el Shell de administración puede ser "transportado" del originador (por ejemplo, el fabricante de componentes eléctricos o diseñador) al sistema de TI a nivel superior (por ejemplo, como un archivo adjunto a un correo electrónico). | representación jerárquica de XML / JSON y formato de archivo de paquete permite diferentes escenarios de transporte.  |
| COMIENZO # 7       | Dependiendo de la naturaleza de los sistemas de nivel superior, puede ser necesario para los objetos de administración para permitir el despliegue de más de un sistema informático de alto nivel.  | Metamodelo y la información de representación independiente de cualquier escenario de implementación.   |
| COMIENZO # 8       | El componente Industrie 4.0, y en particular la Shell Administración, su funcionalidad inherente y los protocolos en cuestión son para ser "encapsulación-capaz" o "separable" de cualquier buses de campo en uso.  | Metamodelo y la información de representación independiente de cualquier escenario de comunicación.   |
| COMIENZO # 9       | El objetivo de la componente Industrie 4.0 es detectar no Industrie 4.0 relaciones de comunicación compatibles de entrada o salida de Shell Administración del objeto y para que sean accesibles a los de extremo a extremo de ingeniería.                                  | No Industrie 4.0 relaciones de comunicación conformes pueden ser modelados por submodelos y por lo tanto ponen a disposición.   |
| COMIENZO # 10      | Debería ser posible asignar lógicamente otros componentes Industrie 4.0 a un 4.0 componente Industrie (por ejemplo, una máquina completa) de una manera tal que hay de anidación (temporal).  | <b>referencias y preparaciones para el Componentes compuestos [ 12] (no incluidos en la parte 1)</b>  |
| COMIENZO # 11      | sistemas de nivel superior deberían poder acceder a todos los componentes 4.0 Industrie de una manera con propósito y limitables, incluso cuando éstos están (temporalmente) lógicamente asignados.   | Alcance de la próxima parte de la serie de documentos; no alcance de esta parte.  |
| COMIENZO # 12      | Características (1) Identificabilidad   | <b>Dada por identificable</b>   |

|               |   |   |
|---------------|---|---|
| COMIENZO # 13 | Características (2) comunicación I4.0 compatible  | No alcance de parte 1   |
| COMIENZO # 14 | Características (3) servicios y I4.0 compatible con la condición múltiple               | La estandarización de submodelos  |
| COMIENZO # 15 | Características (4) Descripción de Virtual  | Disponible por representación virtual ( submodelo y SubmodelElements)                               |
| COMIENZO # 16 | Características semántica (5) I4.0 conformes  | hasSemantics  |
| COMIENZO # 17 | Características (6) Seguridad y protección  | Seguridad por atributo base y Función de acceso de la base. La seguridad no alcance de la parte 1   |
| COMIENZO # 18 | Características (7) Calidad de los servicios  | Metamodelo y la información de representación independiente de cualquier escenario de comunicación. |
| COMIENZO # 19 | Características (8) Estado  | La estandarización de submodelos  |
| COMIENZO # 20 | Características (9) encajamiento  | Los preparativos para Componentes compuestos [ 12] (no incluidos en la parte 1)                     |
| COMIENZO # 21 | La infraestructura mínima debe satisfacer los principios de seguridad por diseño (SBD). | Seguridad por atributo base y Función de acceso de la base.   |

Los siguientes requisitos se basarán en el documento “Estructura de la Administración de Shell: perspectivas trilaterales de

Francia, Italia y Alemania” [18]. Están marcados “Taas”.

Nota: Se utilizó el término “propiedad” en un sentido muy amplio en publicaciones anteriores de la Plataforma Industrie 4.0.

El metamodelo en este documento distingue entre las propiedades en un sentido más clásico como elemento de datos como “temperatura máxima” y otros elementos de submodelos como operaciones, eventos, etc.

| Fuente    | Requisito   | Rastreo  |
|-----------|---|--|
| tAAS- # 1 | El Shell Administración aceptará propiedades de diferentes dominios técnicos en submodelos mutuamente distintos que pueden ser versión controlada y mantenida de forma independiente el uno del otro. | <p>Identificable</p> <p>AdministrativeInformation</p> <p>submodelo</p> <p>Requisitos tAAS- # 1 contiene implícitamente los requisitos de control de versiones.<br/><b>Versiones es compatible con todos los elementos que heredan de Identificable.</b></p> <p>Requisito tAAS- # 1 se cumple debido a que varios submodelos por AAS son posibles. Cada sub-modelo es identificable y una <b>identificable puede contener información administrativa ( Información administrativa) control</b> de versiones. La razón de submodelos para ser identificables es que pueden mantenerse independientemente de otros submodelos (Requisito tAAS- # 1) y que pueden ser reutilizados dentro de diferentes AAS. Sin embargo, ya que los elementos de submodelos pueden referirse a elementos de otras dependencias AAS tienen que ser considerados en el desarrollo paralelo y antes de su reutilización.</p> |
| tAAS- # 2 | El Shell de administración debe ser capaz de incluir las propiedades de una amplia gama de ámbitos técnicos y de [sic] identificar qué dominio se derivan de.   | <p>hasSemantics</p> <p>A través de referencias semánticas definiciones de propiedades de diferentes diccionarios y por lo tanto diferentes dominios pueden ser utilizados dentro de submodelos.</p>  |

|           |  |   |
|-----------|--|---|
|           |  | El único que se requiere es que el dominio de una propiedad se deriva de tiene un <b>identificador único ( semanticId)</b> .  |
| tAAS- # 3 | Para encontrar las definiciones dentro de cada dominio técnico pertinente, diferentes modelos de procedimiento se debe permitir que cumplan, respectivamente, los requisitos de las normas, especificaciones del consorcio y Especificaciones del fabricante conjuntos.  | <p>HasSemantics.semanticId (ver tAAS- # 2)</p> <p>ConceptDescription</p> <p>fabricante patentada propiedad específica - o más en general - descripciones de conceptos o <b>copias de los diccionarios externos son compatibles con la definición ConceptDescriptions. Se hace referencia en semanticId a través de su mundial carné de identidad.</b></p> <p><b>Hasta ahora sólo hay una plantilla de especificación de datos predefinida para Propiedad elementos ( DataSpecification_IEC61360).</b></p> <p>No se recomienda el uso de descripciones de conceptos de propiedad, porque entonces la interoperabilidad no se puede garantizar.</p> |
| tAAS- # 4 | Diferentes depósitos de la Administración en relación con un activo deben ser capaces de hacer referencia a unos de otros. En particular, los elementos de una cáscara de Administración deben ser capaces de desempeñar el papel de una "copia" de los componentes correspondientes de otro Shell Administración. | <p>AssetAdministrationShell.derivedFrom</p> <p><b>los derivado de relación está especialmente diseñado para el apoyo a la relación</b> entre un Shell Administración activo que representa un tipo de activo y las cáscaras de administración activo que representa los casos activos de este tipo de activos.</p> <p>Ver también tAAS- # 16</p>  |
| tAAS- # 5 | Los depósitos de la Administración individuales deben, al tiempo que conserva su estructura, pueden combinar en una cáscara Administración general.  | Los depósitos de la Administración compuestos se tratarán en futuras versiones o partes de la serie de documentos   |
| tAAS- # 6 | La identificación de activos, depósitos de administración, propiedades y relaciones se consigue utilizando un conjunto limitado de identificadores (IRDI, URI y GUID), que proporciona la medida de lo posible oferta singularidad global.   | <p>Identification.idType</p> <p>identificable</p> <p>Requisito tAAS- # 6 se cumple para todos los elementos que heredan de <b>Identificable. Por ejemplo, este es el caso de Activo, AssetAdministrationShell y descripciones</b> de conceptos. Sin embargo, las propiedades (como cualquier otro elemento submodelo) sólo son referibles. Sin embargo, la referencia única es posible a través de la identificación única y <b>el submodelo Referencia vía Llave s concepto. Los tipos de ID soportados incluyen IRDI, URI y GUID (= Personalizado)</b></p> <p>de acuerdo a lo pedido.</p>   |
| tAAS- # 7 | El Shell Administración debe permitir la recuperación de los identificadores alternativos, tales como un identificador de GS1 y GTIN en retorno a la ID de elemento (referencing).   | <p>Asset.assetIdentificationModel</p> <p>Cada activo tiene un identificador único global. Además de este identificador global identificadores locales adicionales pueden especificarse dentro de un sub-modelo especial <b>llamado " assetIdentificationModel ".</b></p> <p>El propio modelo de identificación activo no está predefinido por el metamodelo. Esto significa que hay la necesidad de definir un submodelo que puede contener identificadores alternativos incluyendo referencias semánticas para conocer el significado del identificador adicional.</p>   |
| tAAS- # 8 | El Shell de administración consta de cabecera y el cuerpo.   | <p>AssetAdministrationShell</p> <p>El Shell de administración de activos no distingue explícitamente entre cabecera y el cuerpo. Sin embargo, el Shell de administración de activos tiene atributos definidos que pertenecen a sí mismo como el mundial</p>   |

|            |   |  |
|------------|---|--|
|            |   | <b>Identificación única ( identificación), información de versión ( Información administrativa), una referencia obligatoria para el activo ( activo) representa etc.</b>   |
| tAAS- # 9  | La cabecera contiene información acerca de la identificación.   | <p>AssetAdministrationShell.asset</p> <p>El Shell de Administración de Activos está representando un activo con un identificador único.</p> <p>Ver también tAAS- # 7 Véase también tAAS- # 13</p>  |
| tAAS- # 10 | El cuerpo contiene la información sobre el activo (s) respectiva.   | <p>AssetAdministrationShell.submodels</p> <p>Todos los submodelos dan información respecto de o relacionados con el activo presentado por la AAS.</p> <p><u>Nota:</u> Un Shell de administración de activos representa exactamente un activo. En caso de Shell Administración de Activos compuesto se representa implícitamente varios activos (véase también tAAS- # 5).</p>  |
| tAAS- # 11 | La información y funcionalidad en el Shell de administración es accesible por medio de una interfaz de programación de aplicaciones estándar (API). | Se tratarán en futuras partes de la serie de documentos  |
| tAAS- # 12 | El Shell de administración tiene un identificador único.  | <p>AssetAdministrationShell.identification.id</p> <p><b>Desde AssetAdministrationShell hereda de identifiable</b></p> <p>Requisito se cumple tAAS- # 12.</p>   |
| tAAS- # 13 | El activo tiene un identificador único.   | <p>Asset.identification.id</p> <p><b>Dado que el activo se hereda de identifiable Requisito se cumple tAAS- # 13.</b></p> <p>Ver también Requisito tAAS- # 7.</p> <p><b>Ya que Activo no contiene atributos específicos obligatorios y sólo es adecuado para los sensores etc., también los activos más complejos, como las instalaciones industriales pueden ser modelados (Requisito tAAS- # 14). La única suposición es que la instalación industrial también tiene un identificador único global.</b></p> <p><u>Nota:</u> Véase también el compuesto de Activos Administración Shell (ver tAAS- # 5) que permite el modelado de los activos complejos constituidos por otros activos que están representados por una AAS cada por sí mismos.</p> |
| tAAS- # 14 | Una instalación industrial es también un activo, tiene una cáscara Administración y es accesible por medio de identificación.                       | <p>activo Asset.identification.id</p> <p><b>Ya que Activo no contiene atributos específicos obligatorios y sólo es adecuado para los sensores etc., también los activos más complejos, como las instalaciones industriales pueden ser modelados. La única suposición es que la instalación industrial también tiene un identificador único global. Nota: Ver también Compuesto Shell AssetAdministration (ver tAAS- # 5) que permite el modelado de los activos complejos constituidos por otros activos que están representados por una AAS cada por sí mismos.</b></p>   |
| tAAS- # 15 | Tipos y casos deben ser identificadas como tales.   | <p><b>HasKind (Con tipo = tipo o clase = Instancia) de activos</b></p> <p>AssetAdministrationShell.derivedFrom</p>   |

|            |  |  |
|------------|--|--|
|            |  | <p>Dado que el activo se hereda de HasKind Requisito tAAS- # 15 se ha cumplido y tipos de activos puede distinguirse de casos activos.</p> <p>Además, una derivado de relación puede establecerse entre el AAS para una instancia de activos y el AAS para el tipo de activos.</p>   |
| tAAS- # 16 | El Shell de administración puede incluir referencias a otras cáscaras administración o información inteligente de fabricación. | <p>Blob Archivo</p> <p>ReferenceElement</p> <p>AssetAdministrationShell.derivedFrom</p> <p>los derivado de relación entre dos AAS es especial y es por ejemplo utilizado para establecer una relación entre las instancias de activos y el tipo de activos.</p> <p>Para AAS compuesto (ver tAAS- # 5) existe también la relación con AAS el AAS compuesto se compone de. los ReferenceElement es muy genérico y se puede hacer referencia a otra AAS, así como información dentro de otro AAS o incluso alguna información que está completamente fuera de cualquier AAS (siempre y cuando se tiene un identificador único global).</p> <p>Los archivos y BLOB se pueden utilizar como elementos de submodelos para incluir información de fabricación muy genérica que no es o no puede ser modelado a través de las propiedades o los otros elementos de submodelos definidos para la Shell Administración de Activos.</p> |
| tAAS- # 17 | Las propiedades adicionales, por ejemplo, específico del fabricante, debe ser posible.   | <p>HasDataSpecification</p> <p>ConceptDictionary</p> <p>A través de plantillas de especificación de datos atributos adicionales para los activos, propiedades y otros elementos de submodelos, submodelos, puntos de vista e incluso el Shell AssetAdministration en sí puede ser definido y comprobado por las herramientas.</p> <p>Nuevas descripciones de las propiedades de propiedad se pueden añadir de forma local al diccionario de conceptos local del AAS y se utilizan para la definición semántica de las propiedades u otros elementos de submodelos. Una extensión del metamodelo mediante la definición de clases de propiedad que heredan de las clases definidas de este metamodelo también es posible.</p> <p>Via API (ver tAAS- # 11) nuevas propiedades, se pueden añadir otros elementos de submodelos y submodelos - asumió el correspondiente acceso se dan permisos.</p>                             |
| tAAS- # 18 | Un número mínimo confiable de propiedades se debe definir para cada Shell Administración.                                      | <p><b>hasKind para submodelo y SubmodelElements</b></p> <p>Un número mínimo fiable de las propiedades se define por la propia metamodelo. Se les llama (clase) atribuye.</p> <p><b>HasKind (Con tipo = Tipo) para submodelo y elemento submodelo s permite la definición de tipos de submodelos (elemento). Estos tipos están referenciados a través semanticId.</b></p> <p><u>Nota:</u> el término propiedad dentro del metamodelo tiene una semántica especiales y no se mezcla con los atributos disponibles de manera implícita de las diferentes clases. A pesar de estos atributos como</p>  |

|            |   |   |
|------------|---|---|
|            |   | bien podría basarse en las normas existentes que hay propiedades en el sentido de que una referencia semántica puede añadir que define la semántica externamente: La semántica se define para el metamodelo sí en las tablas de clase dentro de este documento.   |
| tAAS- # 19 | Las propiedades y otros elementos de información en el Shell de administración deben ser adecuados para tipos e instancias. | <p><b>HasKind (Con tipo = tipo o clase = instancia) para submodelo y SubmodelElement</b></p> <p>Todos los elementos que heredan de HasKind puede distinguir entre tipos e instancias. Esto es especialmente cierto para las SubmodelElement y Submodelo.</p> <p><b>Nota:</b> Los submodelos o propiedades de tipo = Tipo no describen un activo de tipo = Tipo. Esto se realiza a través de propiedades de = tipo de instancia.</p>   |
| tAAS- # 20 | Tiene que haber una capacidad de estructuración jerárquica y contable de las propiedades.                                   | <p>DataElementCollection</p> <p>Requisito tAAS- # 20 se cumple por las colecciones de elementos de datos. La colección puede ser caracteriza además si se ordena y si puede contener duplicados. Las colecciones se construyen de forma recursiva y por lo tanto contienen otros elementos submodelo de la misma AAS. Para hacer referencia a propiedades o otros elementos de submodelos de otro AAS una referencia ( ReferenceElement) o elemento de relación ( RelationshipElement) debe incluirse como parte de la propiedad compleja.</p>  |
| tAAS- # 21 | Propiedades podrán hacer referencia a otras propiedades, incluso en otras cáscaras de administración.                       | <p>DataElementCollection</p> <p>ReferenceElement RelationshipElement</p> <p><b>OperationVariable en Operación</b></p> <p>Un elemento de referencia puede hacer referencia a cualquiera de cualquier otro elemento que es referible (es decir, que hereda de Referible) dentro de la misma o de otra AAS. O se puede hacer referencia a las entidades completamente fuera de cualquier AAS a través de su identificación global.</p> <p><b>Nota:</b> Para los elementos dentro de los mismos AAS referencia a que no siempre es necesario el uso de una propiedad de referencia. Dependiendo del contexto también colecciones de elementos de submodelos, relaciones, etc. podrían ser más adecuados.</p> <p><b>Dentro operación s también otros elementos (que deben tener su propia referencia semántica, OperationVariable) se hace referencia o se utiliza como entrada o argumento de salida.</b></p> |
| tAAS- # 22 | Las propiedades deben ser capaces de hacer referencia a la información y funciones de la Administración de Shell.           | <p>Operación</p> <p>Ver también tAAS- # 21</p> <p>Funciones en el sentido de entidades ejecutables se representan como <b>operación s.</b></p>  |

Los siguientes requisitos se han obtenido a partir del documento "Seguridad de la Concha Administrativo" [14]. Están marcados como "SecAAS"

| CARNÉ DE IDENTIDAD | Requisito | Rastreo |
|--------------------|-----------|---------|
|--------------------|-----------|---------|

|             |   |  |
|-------------|---|--|
| SecAAS- # 1 | Identificación y autenticación: Se debe asegurar que las entidades correctas (Shell y usuarios de administración) interactúan entre sí. Esto se aplica tanto en un contexto de comunicación local (dentro de una máquina o instalación), y en un contexto global (a través de empresas). La identificación clara (por autenticación) de los interlocutores de la comunicación es un requisito básico para la interacción con una cáscara de gestión. Sin ellos, las características de seguridad adicionales (confidencialidad, integridad, etc.) no pueden ser garantizadas. | <p>Certificado</p> <p>Security.trustAnchor</p> <p>anclas de confianza se realizan mediante certificados.</p> <p><u>la gestión de certificados se detallará en partes o versiones del documento (series) futuras.</u></p>   |
| SecAAS- # 2 | De usuario y derechos de gestión: una cáscara AssetAdministration puede tener diferentes compañeros de interacción. Para controlar las posibilidades de interacción con el Shell de administración, una administración de usuarios y derechos es necesario.   | <p>Security.policyAdministrationPoint</p> <p>AccessControl</p> <p>AccessControl.accessPermissionRule</p> <p>No hay un manejo sujeto explícito en el AAS: Se supone que la identidad del sujeto que solicita el acceso con un rol determinado (a través de la API - ver tAAS- # 11) se autentica fuera del AAS. El AAS puede comprobar la autorización a través del punto final al tema atribuye proveedor.</p> <p>Para cada objeto en las reglas de permiso de acceso Shell de administración de activos puede ser definido.</p> |
| SecAAS- # 3 | Comunicación segura: La comunicación con el Shell de administración podrá incluir información sensible. Del mismo modo, un cambio en la comunicación entre la Administración de Shell y sus socios de comunicación puede causar trastornos graves y peligrosas de una máquina o instalación. Por tanto, es obligatorio que se tomen las medidas adecuadas para garantizar la seguridad de las comunicaciones. Esto debe hacerse mediante el uso de protocolos de seguridad apropiadas.  | No aplica  |
| SecAAS- # 4 | El registro de eventos: La trazabilidad de la interacción con el Shell de administración juega un papel crucial en la detección de incidentes de seguridad. Esta trazabilidad se logra a través de la tala de registro / evento y auditoría. por lo tanto, el Shell de administración debe proporcionar métodos que registran accesos y cambios en el estado de la cáscara de gestión sin modificación. También es importante ser capaz de recoger de forma centralizada y evaluar esta información del evento.   | manejo de la historia se detallará en partes o versiones del documento (serie) futuros.  |



## Anexo B. Las plantillas para las Tablas UML

En este anexo, se explican las plantillas utilizadas para la especificación de elemento.

Plantilla de documento clases (elementos):

|                            |             |      |      |          |
|----------------------------|-------------|------|------|----------|
| Clase:                     |             |      |      |          |
| Explicación:               |             |      |      |          |
| Hereda de:                 |             |      |      |          |
| Atributo (* = obligatorio) | Explicación | Tipo | Tipo | Tarjeta. |
|                            |             |      |      |          |
|                            |             |      |      |          |
|                            |             |      |      |          |

Kind se define con la semántica de UML:

- attr: atributo (tipo no es un objeto)
- aggr: agregación (no existe independiente de su padre)
- ref: composición (existe independiente de su matriz) Además, hay

una especie:

- ref \*: referencia a través de la clase "de referencia" con target = <tipo>
- Para obtener más información sobre la referencia a la cláusula 3.6.15 Ver tarjeta. es la

cardinalidad.

Plantilla para enumeraciones:

|              |             |
|--------------|-------------|
| Enumeración: |             |
| Explicación: |             |
| Literal      | Explicación |
|              |             |

## Anexo C. Leyenda para el modelado UML

Figura 59 Agregación en Metamodel en UML - Leyenda

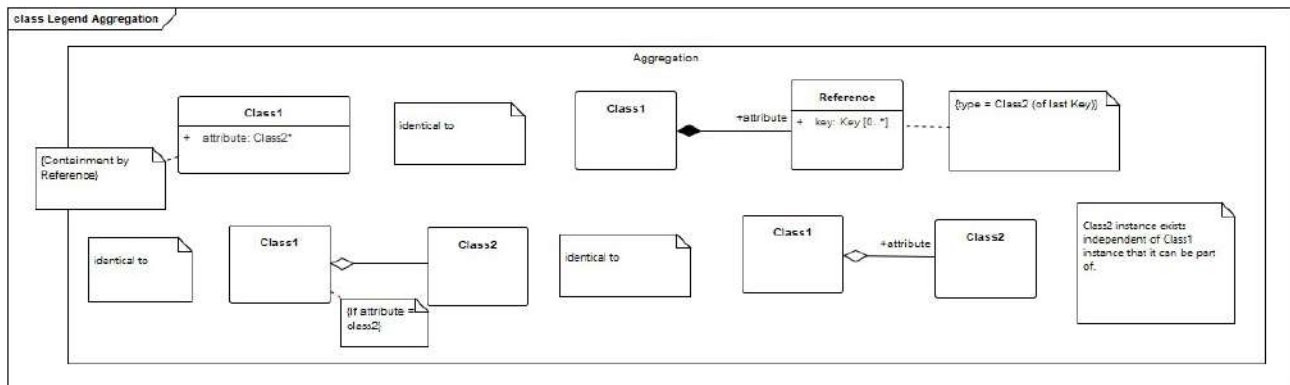


Figura 60 Asociación en Metamodel en UML - Leyenda

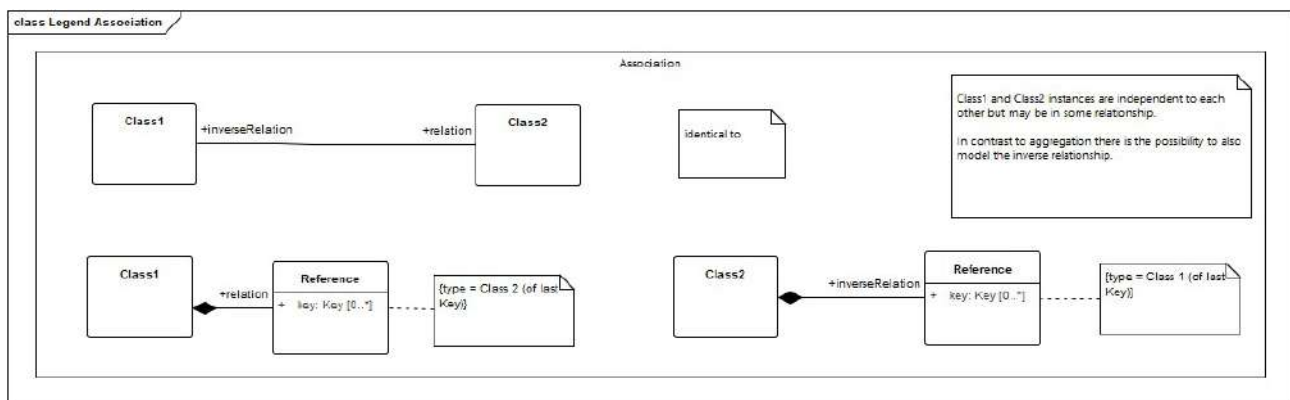


Figura 61 Composición en Metamodel en UML - Leyenda

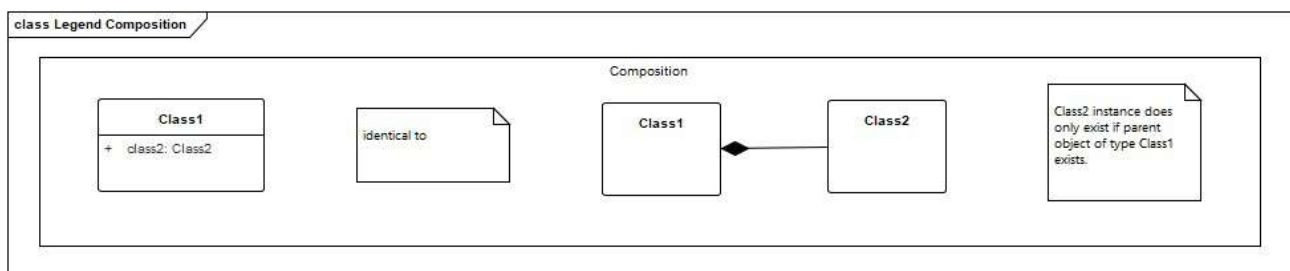


Figura 62 Identificación en Metamodel en UML - Leyenda

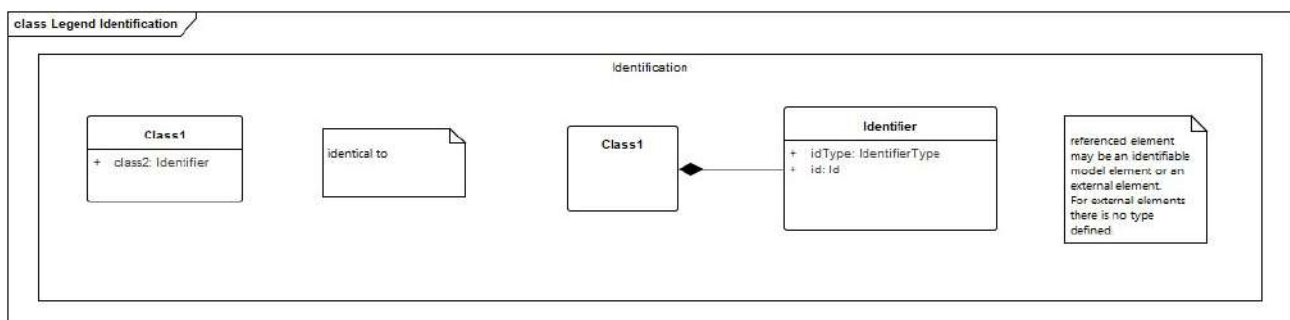


Figura 63 Clases herencia en Metamodel en UML - Leyenda

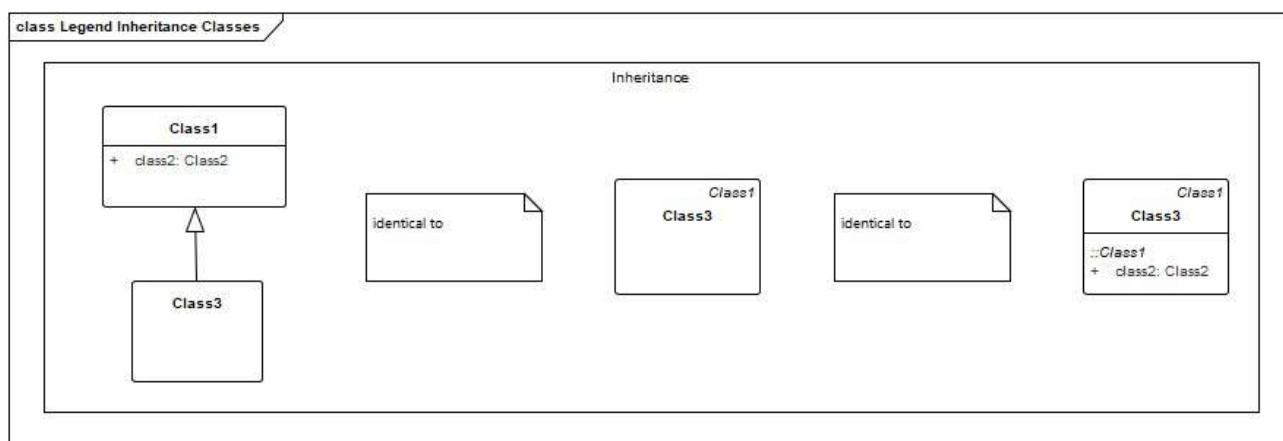
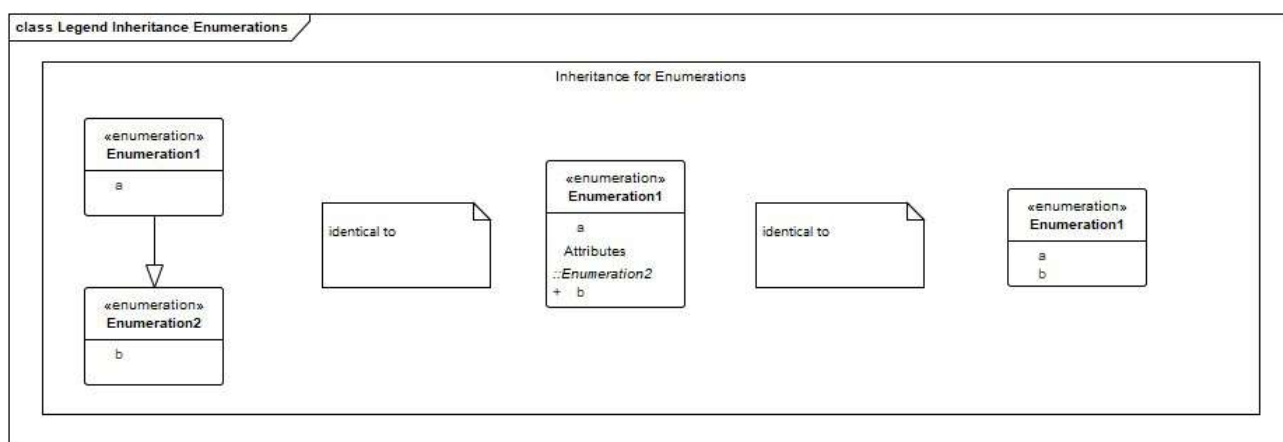


Figura 64 Las enumeraciones herencia en Metamodel en UML - Leyenda

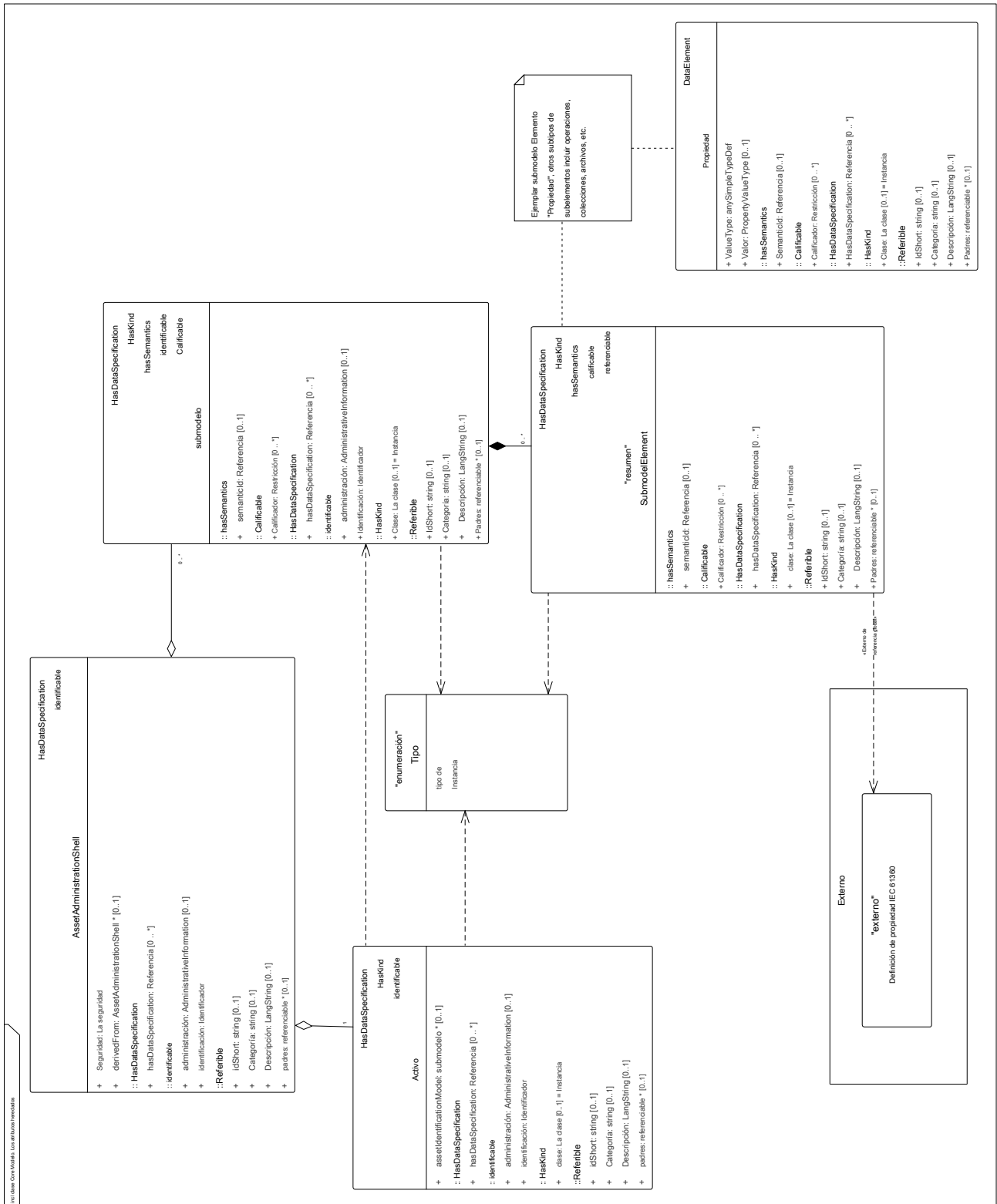


# Anexo D. UML Metamodel con heredad

## atributos

En este anexo algunos diagramas UML se muestran junto con todos los atributos inerted. Ver también Figura 29 para

un diagrama con todo heredado atributos de ConceptDescription. Figura Modelo 65 Core con atributos heredados



La Figura 66 de control de acceso con atributos heredados

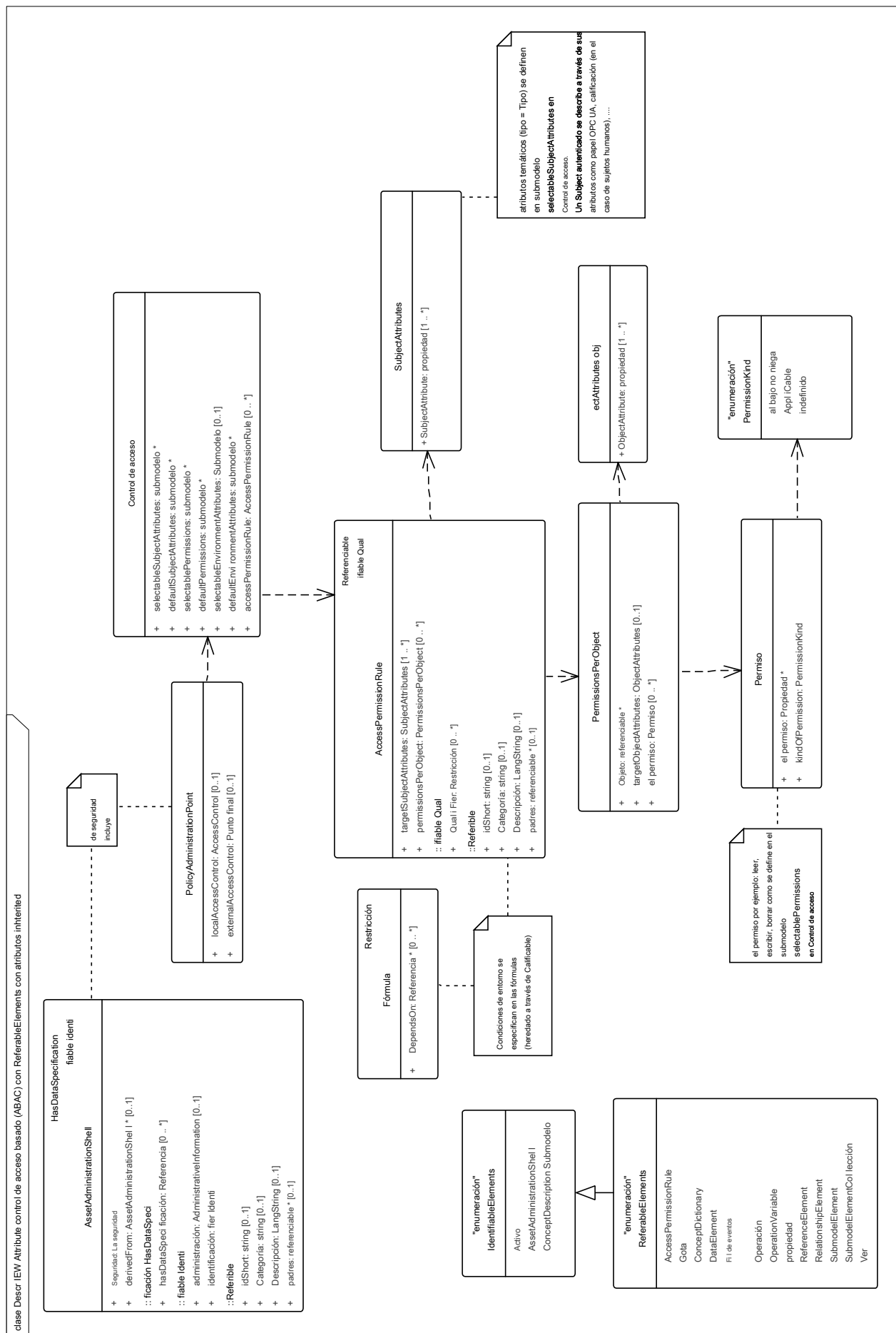
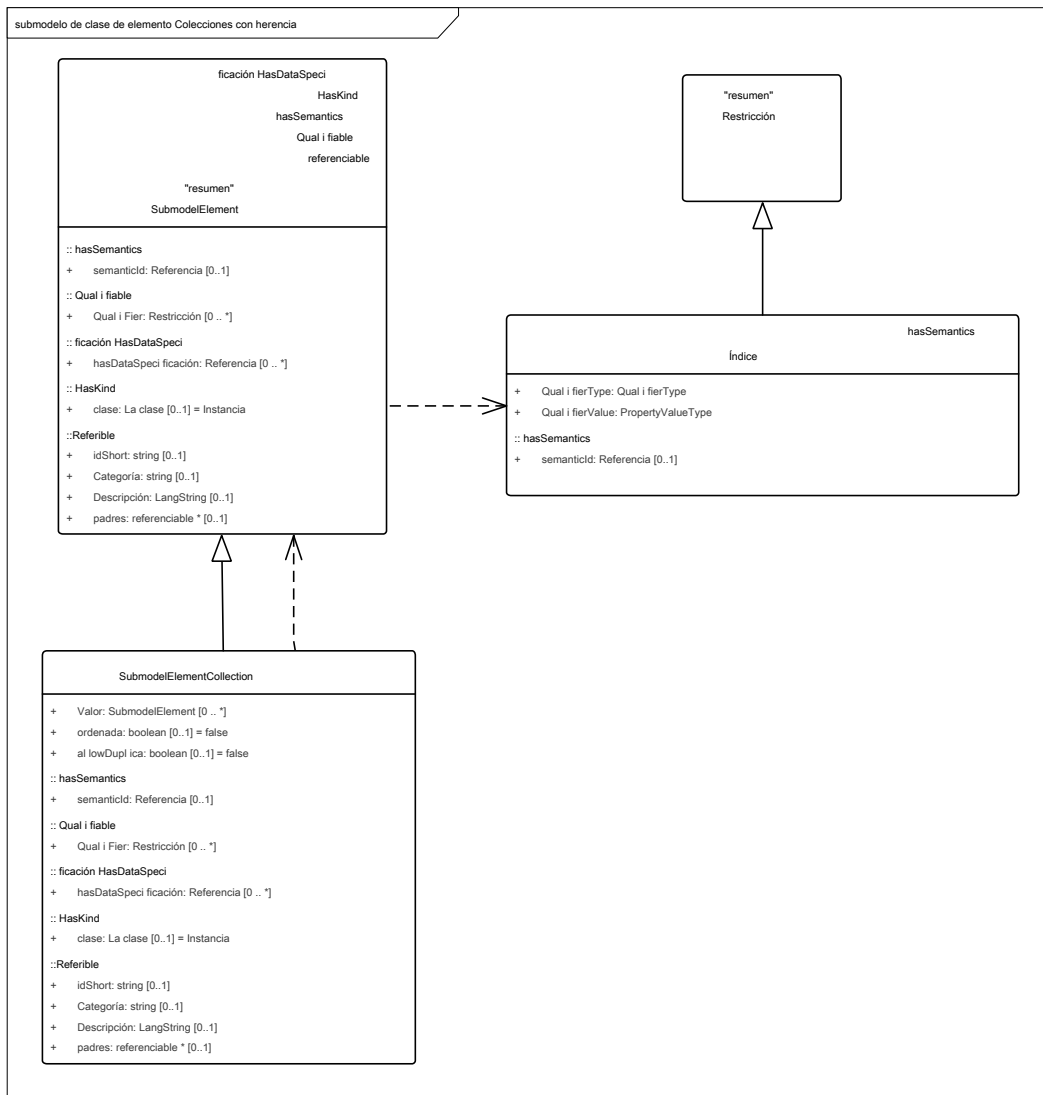


Figura 67 Submodelo Collection Element con la herencia



## Anexo E. esquemas XML y ejemplo completo

### 1. Los esquemas XML para Shell Administración

El esquema está dividido en dos partes:

- Los principales conceptos de la Concha de Administración (AAS.xsd)
- La Especificación de la plantilla de datos IEC61360 (IEC61360.xsd)

Posteriormente, se discute un ejemplo en XML.

### 2. Esquema para la administración general de Shell

```
<?xml versión = "1.0" codificación = "UTF-8" ?>
<esquema targetNamespace = "http://www.admin-shell.io/aas/1/0"
  elementFormDefault = "calificado" xmlns = "http://www.w3.org/2001/XMLSchema"
  xmlns:AAS = "http://www.admin-shell.io/aas/1/0" xmlns:IEC61360 = "http://www.admin-shell.io/IEC61360/1/0" > <import schemaLocation = "IEC61360.xsd" espacio de nombres = "http://www.admin-shell.io/IEC61360/1/0"
  >
  </import> <elemento nombre = "Aasenv" tipo = "Aas: aasenv" /> </Element>

  <complexType nombre = "Aasenv" /> <Secuencia>

    <elemento nombre = "assetAdministrationShells" tipo = "Aas: assetAdministrationShells" *
      minOccurs = "0" maxOccurs = "1" /> <element nombre = "bienes" tipo = "Aas: assets" minOccurs = "0" maxOccurs = "1" /> <element nombre
      = "submodelos" tipo = "Aas: submodels" minOccurs = "0" maxOccurs = "1" /> <element nombre = "conceptDescriptions" tipo = "Aas: conceptDescriptions" />

    minOccurs = "0" maxOccurs = "1" /> </Element>
  </ Secuencia> </
  complexType>

  <complexType nombre = "AssetAdministrationShells" /> <Secuencia>

    <elemento nombre = "AssetAdministrationShell" tipo = "Aas: assetAdministrationShell" /
      minOccurs = "0" maxOccurs = "ilimitado" />
    </ Element> </
    secuencia> </
    complexType>

  <complexType nombre = "Assets" /> <Secuencia>

    <elemento nombre = "activo" tipo = "Aas: asset" minOccurs = "0"
      maxOccurs = "ilimitado" />
    </ Element> </
    secuencia> </
    complexType>

  <complexType nombre = "Asset" /> <Secuencia>

    <grupo arbitrario = "Aas: identificable" /> </ Grupo> <grupo arbitrario = "aas: hasDataSpecifications" /> </ Grupo> <grupo arbitrario = "Aas: hasKind" /> </ Grupo> <elemento nombre
      = "AssetIdentificationModelRef" tipo = "Aas: reference" minOccurs = "0"
      maxOccurs = "1" /> </ Element> </
      secuencia> </ complexType>

  <complexType nombre = "AssetAdministrationShell" /> <Secuencia>

    <grupo arbitrario = "Aas: identificable" /> </ Grupo> <grupo arbitrario = "aas: hasDataSpecifications" /> </ Grupo> <elemento nombre = "derivado de" tipo = "Aas: reference" minOccurs = "0" maxOccurs
      = "1" /> </ Element> <element nombre = "AssetRef" tipo = "Aas: reference" minOccurs = "1" maxOccurs = "1" /> </ Element> <element nombre = "submodelRefs" tipo = "Aas:
      submodelRefs" minOccurs = "0" maxOccurs = "1" /> </ Element> <element nombre = "puntos de vista" tipo = "Aas: views" minOccurs = "0" maxOccurs = "1" /> </ Element>
    <element nombre = "conceptDictionaries"
      tipo = "Aas: conceptDictionaries" minOccurs = "0" maxOccurs = "1" />
    </ Element> </
    secuencia> </
    complexType>

  <complexType nombre = "Submodel" /> <Secuencia> <grupo arbitrario = "Aas: identificable" /> </ Grupo> <grupo arbitrario = "aas: hasDataSpecifications" /> </
  Grupo> <grupo arbitrario = "aas: hasSemantics" /> </ Grupo> <grupo arbitrario = "Aas: hasKind" /> </ Grupo> <grupo arbitrario = "Aas: calificable" /> </ Grupo>
  <elemento nombre = "submodelElements" tipo = "Aas: submodelElements" /> </ Element>
```

```

</ Secuencia> </
complexType>

<complexType nombre = "ConceptDescription_f" > <Secuencia>

    <grupo árbitro = "Ass: identifiable" > </ Grupo> <grupo árbitro = "ass: hasDataSpecifications" > </ Grupo> <elemento nombre = "IsCaseOf" tipo = "Ass: reference_f" maxOcurrencias = "ilimitado" minOcurrencias
    = "0" > </ Element>

    </ Secuencia> </
complexType>

<complexType nombre = "View_f" > <Secuencia>

    <grupo árbitro = "Ass: referable" > </ Grupo> <grupo árbitro = "ass: hasSemantics" > </ Grupo> <grupo árbitro = "ass: hasDataSpecifications" > </ Grupo>
    <elemento nombre = "containedElements" tipo = "Ass: containedElements_f" > </ Element>

    </ Secuencia> </
complexType>

<complexType nombre = "SubmodelElements_f" > <Secuencia>

    <elemento nombre = "SubmodelElement" tipo = "Ass: submodelElement_f" •
    minOcurrencias = "0" maxOcurrencias = "ilimitado" > </ Element>

    </ Secuencia> </
complexType>

<complexType nombre = "SubmodelElementAbstract_f" > <Secuencia> <grupo árbitro = "Ass:
referable" > </ Grupo> <grupo árbitro = "ass: hasSemantics" > </ Grupo> <grupo árbitro = "ass:
hasDataSpecifications" > </ Grupo> <grupo árbitro = "Ass: hasKind" > </ Grupo> <grupo árbitro =
"Ass: calificable" > </ Grupo>

    </ Secuencia> </
complexType>

<complexType nombre = "SubmodelRefs_f" > <Secuencia>

    <elemento nombre = "SubmodelRef" tipo = "Ass: reference_f" minOcurrencias = "0" maxOcurrencias = "ilimitado" > </ Element>

    </ Secuencia> </
complexType>

<complexType nombre = "Views_f" > <Secuencia>

    <elemento nombre = "ver" tipo = "Ass: view_f" minOcurrencias = "0" maxOcurrencias = "ilimitado" > </ Element>

    </ Secuencia> </
complexType>

<complexType nombre = "ConceptDictionary_f" > <Secuencia> <grupo árbitro = "Ass: referable" > </ Grupo> <elemento nombre = "conceptDescriptionRefs" tipo = "Ass:
conceptDescriptionsRef_f" •

    minOcurrencias = "0" maxOcurrencias = "1" > </ Element>

    </ Secuencia> </
complexType>

<complexType nombre = "ConceptDescriptions_f" > <Secuencia>

    <elemento nombre = "ConceptDescription" tipo = "Ass: conceptDescription_f" •
    minOcurrencias = "0" maxOcurrencias = "ilimitado" > </ Element>

    </ Secuencia> </
complexType>

<complexType nombre = "ConceptDictionaries_f" > <Secuencia>

    <elemento nombre = "ConceptDictionary" tipo = "Ass: conceptDictionary_f" •
    minOcurrencias = "0" maxOcurrencias = "ilimitado" > </ Element>

    </ Secuencia> </
complexType>

<complexType nombre = "Submodels_f" > <Secuencia>

    <elemento nombre = "Submodelo" tipo = "Ass: submodel_f" minOcurrencias = "0" maxOcurrencias = "ilimitado" > </ Element>

    </ Secuencia> </
complexType>

<complexType nombre = "ContainedElements_f" > <Secuencia>

    <elemento nombre = "ContainedElementRef" tipo = "Ass: reference_f" > </ Element>

    </ Secuencia> </
complexType>

<complexType nombre = "SubmodelElement_f" > <Opción>

    <elemento nombre = "propiedad" tipo = "Ass: property_f" > </ Element> <elemento nombre = "expediente" tipo = "Ass:
file_f" > </ Element> <elemento nombre = "gota" tipo = "Ass: blob_f" > </ Element> <elemento nombre = "ReferenceElement"

    tipo = "Ass: referenceElement_f" >

    </ Element>

```



```

<elemento nombre = "SubmodelElementCollection"
  tipo = "Aas: submodelElementCollection_f" >
</ Element> <element nombre = "RelationshipElement"

  tipo = "Aas: relationshipElement_f" >
</ Element> <element nombre = "operación" tipo = "Aas: operation_f" > </ Element> <element nombre = "OperationVariable"

  tipo = "Aas: operationVariable_f" >
</ Element> <element nombre = "evento" tipo = "Aas: event_f" > </ Element>

</ Choice> </
complexType>

<complexType nombre = "Property_f" > <ComplexContent>

  <extensión base = "Aas: submodelElementAbstract_f" > <Secuencia>

    <elemento nombre = "tipo de valor" tipo = "cadena" > </ Element> <element nombre = "valor" tipo = "Aas: propertyValueType_f" maxOccurs = "1" minOccurs = "0" > </ Element>
    <element nombre = "ValueId" tipo = "Aas: reference_f" maxOccurs = "1" minOccurs = "0" > </ Element>

  </ Secuencia> </
extension> </ complexContent>
</ complexType>

<complexType nombre = "file" > <ComplexContent>

  <extensión base = "Aas: submodelElementAbstract_f" > <Secuencia> <elemento nombre = "tipo MIME" tipo = "cadena"
  > </ Element> <element nombre = "valor" tipo = "Aas: pathType_f" > </ Element> </ secuencia> </ extension> </
complexContent> </ complexType>

<complexType nombre = "Blob_f" > <ComplexContent>

  <extensión base = "Aas: submodelElementAbstract_f" > <Secuencia> <elemento nombre = "tipo MIME" tipo = "cadena"
  > </ Element> <element nombre = "valor" tipo = "Aas: blobType_f" > </ Element> </ secuencia> </ extension> </
complexContent> </ complexType>

<complexType nombre = "ReferenceElement_f" > <ComplexContent>

  <extensión base = "Aas: submodelElementAbstract_f" > <Secuencia> <elemento nombre = "valor" tipo = "Aas:
reference_f" > </ Element> </ secuencia> </ extension> </ complexContent> </ complexType>

<complexType nombre = "SubmodelElementCollection_f" > <ComplexContent>

  <extensión base = "Aas: submodelElementAbstract_f" > <Secuencia> <elemento nombre = "valor" tipo = "Aas: submodelElements_f"
  </ Element> <element nombre = "ordenado" tipo = "Booleano" > </ Element> <element nombre = "AllowDuplicates" tipo = "Booleano"
  </ Element> </ secuencia> </ extension> </ complexContent> </ complexType>

<complexType nombre = "RelationshipElement_f" > <ComplexContent>

  <extensión base = "Aas: submodelElementAbstract_f" > <Secuencia> <elemento nombre = "primero" tipo = "Aas:
reference_f" > </ Element> <element nombre = "segundo" tipo = "Aas: reference_f" > </ Element> </ secuencia> </
extension> </ complexContent> </ complexType>

<complexType nombre = "Operation_f" > <ComplexContent>

  <extensión base = "Aas: submodelElementAbstract_f" > <Secuencia> <elemento nombre = "en" tipo = "Aas: operationVariable_f"
  </ Element> <element nombre = "afuera" tipo = "Aas: operationVariable_f" > </ Element> </ secuencia> </ extension> </
complexContent>

```

```

</ ComplexType>

<complexType nombre = "OperationVariable_f" > <ComplexContent>

    <extensión base = "Aas: submodelElementAbstract_f" > <Secuencia> <elemento nombre = "valor" tipo = "Aas:
    submodelElement_f" > </ Element> </ secuencia> </ extension> </ complexContent> </ complexType>

<complexType nombre = "Event_f" > <ComplexContent>

    <extensión base = "Aas: submodelElementAbstract_f" > <Secuencia> </ secuencia> </
    extension> </ complexContent> </ complexType>

<complexType nombre = "DataSpecificationContent_f" > <Opción>

    <elemento nombre = "DataSpecificationIEC61360" tipo = "IEC61360: dataSpecificationIEC61360_f" > </ Element>
    </ Choice> </
    complexType>

<complexType nombre = "ConceptDescriptionsRef_f" > <Secuencia>

    <elemento nombre = "ConceptDescriptionRef" tipo = "Aas: reference_f" *
    minOccurs = "0" maxOccurs = "ilimitado" > </ Element>
    </ Secuencia> </
    complexType>

<complexType nombre = "PathType_f" > <SimpleContent>

    <extensión base = "cadena" > </ Extension>
    </ SimpleContent> </
    complexType>

<complexType nombre = "BlobType_f" > <SimpleContent>

    <extensión base = "Base64Binary" > </ Extension>
    </ SimpleContent> </
    complexType>

<complexType nombre = "IdPropertyDefinition_f" >
<SimpleContent>
    <extensión base = "cadena" > <Atributo nombre = "tipo de identificación" tipo = "cadena" />
    </ Extension> </ simpleContent> </ complexType>

<complexType nombre = "IdShort_f" > <SimpleContent>

    <extensión base = "cadena" > </ Extension> </
    simpleContent> </ complexType>

<complexType nombre = "Administration_f" > <Secuencia>

    <elemento nombre = "versión" tipo = "cadena" minOccurs = "0"
    maxOccurs = "1" /> <Elemento nombre = "revisión" tipo = "cadena" minOccurs = "0"
    maxOccurs = "1" /> </
    Secuencia> </ complexType>

<complexType nombre = "Identification_f" > <SimpleContent>

    <extensión base = "cadena" > <Atributo nombre = "tipo de identificación" utilizar = "Optional"
    > <SimpleType>

        <restricción base = "cadena" > <Enumeración valor = "URI" > </ Enumeración>
        <enumeración valor = "IRDI" > </ Enumeración> <enumeración valor = "Personalizado" >
        </ Enumeración>

        </ Restricción> </
        simpleType> </ attribute> </ extension>
    </ simpleContent> </ complexType>

<grupo nombre = "Identifiable" > <Secuencia>

```

```

<grupo árbitro = "Ass: referable" > </ Grupo> <elemento nombre = "identificación" tipo = "Ass: identification_f" •
    minOcurrencias = "0" maxOcurrencias = "1" > </ Element> <element nombre = "administración" tipo = "Ass: administration_f" •
    minOcurrencias = "0" maxOcurrencias = "1" > </ Element>
</ Secuencia> </
grupo>

<grupo nombre = "referible" > <Secuencia>
    <elemento nombre = "IdShort" tipo = "Ass: idShort_f" minOcurrencias = "0" maxOcurrencias = "1" > </ Element> <element nombre = "categoría" tipo = "cuerda" minOcurrencias = "0"
    maxOcurrencias = "1" > </ Element> <element nombre = "descripción"
        tipo = "Ass: langStrings_f" minOcurrencias = "0" maxOcurrencias = "1" >
    </ Element> <element nombre = "padre" tipo = "cuerda" minOcurrencias = "0" maxOcurrencias = "1" > </ Element>

</ Secuencia> </
grupo>

<complexType nombre = "Qualifiers_f" > <Secuencia>
    <elemento nombre = "Indice" tipo = "cuerda" minOcurrencias = "0" maxOcurrencias = "ilimitado" > </ Element>
</ Secuencia> </
complexType>

<grupo nombre = "Calificable" > <Secuencia>
    <elemento nombre = "Indice" tipo = "Ass: constraint_f" minOcurrencias = "0" maxOcurrencias = "1" > </ Element>
</ Secuencia> </
grupo>

<grupo nombre = "hasDataSpecifications" > <Secuencia>
    <elemento nombre = "EmbeddedDataSpecification" tipo = "Ass: embeddedDataSpecification_f" •
    maxOcurrencias = "ilimitado" minOcurrencias = "0" > </ Element>
</ Secuencia> </
grupo>

<grupo nombre = "hasSemantics" > <Secuencia>
    <elemento nombre = "Semantickd" tipo = "Ass: semantickd_f" minOcurrencias = "0" > </ Element>
</ Secuencia> </
grupo>

<complexType nombre = "Semantickd_f" > <ComplexContent>
    <extensión base = "Ass: reference_f" > </ Extension>
</ ComplexContent> </
complexType>

<complexType nombre = "Reference_f" > <Secuencia>
    <elemento nombre = "llaves" tipo = "Ass: keys_f" > </ Element>
</ Secuencia> </
complexType>

<complexType nombre = "Qualifier_f" > <Secuencia>
    <grupo árbitro = "ass: hasSemantics" > </ Grupo> <elemento nombre = "QualifierType" tipo = "cuerda" > </ Element> <element nombre = "QualifierValue" tipo = "cuerda" maxOcurrencias = "1" minOcurrencias
    = "0" > </ Element> <element nombre = "QualifierValueId" tipo = "Ass: reference_f" maxOcurrencias = "1" minOcurrencias = "0" > </ Element>

</ Secuencia> </
complexType>

<complexType nombre = "Formula_f" > <Secuencia>
    <elemento nombre = "depende de" tipo = "Ass: references_f" > </ Element>
</ Secuencia> </
complexType>

<complexType nombre = "Constraint_f" > <Opción>
    <elemento nombre = "Indice" tipo = "Ass: qualifier_f" > </ Element> <element nombre = "fórmula" tipo = "Ass: formula_f" > </
    Element>
</ Choice> </
complexType>

<complexType nombre = "References_f" > <Secuencia>
    <elemento nombre = "referencia" tipo = "Ass: reference_f" minOcurrencias = "0" maxOcurrencias = "ilimitado" > </ Element>
</ Secuencia> </
complexType>

<grupo nombre = "HasKind" > <Secuencia>
    <elemento nombre = "tipo" minOcurrencias = "0" maxOcurrencias = "1" > <SimpleType>
        <restricción base = "cuerda" > <Enumeración valor = "Tipo" > </ Enumeración> <enumeración valor
        = "Ejemplo" > </ Enumeración>
    </ SimpleType>
</ Secuencia>

```

```

    </ Restricción> </
    simpleType> </ element> </
    secuencia> </ grupo>

<complexType nombre = "Keys_f" > <Secuencia>

    <elemento árbitro = "Aas: tecla" minOccurs = "0" maxOccurs = "ilimitado" > </ Element>
    </ Secuencia> </
    complexType>

<elemento nombre = "llave" tipo = "Aas: key_f" > </ Element>

<attributeGroup nombre = "keyTypes" > <Atributo nombre = "LocalKeyType" utilizar = "Opcional" >
    <SimpleType>

        <restricción base = "cadena" > <Enumeración valor = "IdShort" > </ Enumeración>

        </ Restricción> </ simpleType> </ attribute> <atributo nombre = "IdentifierType" utilizar
        = "Opcional" > <SimpleType>

            <restricción base = "cadena" > <Enumeración valor = "IRDI" > </ Enumeración>
            <enumeración valor = "URI" > </ Enumeración> <enumeración valor = "Personalizado" >
            </ Enumeración>

        </ Restricción> </
        simpleType> </ attribute> </
        attributeGroup>

<complexType nombre = "Key_f" > <SimpleContent>

    <extensión base = "cadena" > <Atributo nombre = "tipo" >
        <SimpleType>

            <restricción base = "cadena" > <Enumeración valor = "GlobalReference" > </ Enumeración> <enumeración valor = "ConceptDictionary"
            > </ Enumeración>

            <enumeración valor = "AccessPermissionRule" > </ Enumeración> <enumeración valor = "DataElement" > </ Enumeración>
            <enumeración valor = "Ver" > </ Enumeración> <enumeración valor = "Propiedad" > </ Enumeración> <enumeración valor = "SubmodelElement"
            > </ Enumeración> <enumeración valor = "Expediente" > </ Enumeración> <enumeración valor = "Gota" > </ Enumeración>
            <enumeración valor = "ReferenceElement" > </ Enumeración> <enumeración valor = "SubmodelElementCollection" > </
            Enumeración> <enumeración valor = "RelationshipElement" > </ Enumeración> <enumeración valor = "Evento" > </
            Enumeración> <enumeración valor = "Operación" > </ Enumeración> <enumeración valor = "OperationVariable" > </
            Enumeración> <enumeración valor = "AssetAdministrationShell" > </ Enumeración> <enumeración valor = "Submodelo" > </
            Enumeración> <enumeración valor = "ConceptDescription" > </ Enumeración> <enumeración valor = "Activo" > </
            Enumeración> </ restricción> </ simpleType> </ attribute>

        </ Enumeración>

    <atributo nombre = "tipo de identificación" >
        <SimpleType>
            <restricción base = "cadena" > <Enumeración valor = "IdShort" > </ Enumeración> <enumeración valor
            = "IRDI" > </ Enumeración> <enumeración valor = "URI" > </ Enumeración> <enumeración valor = "Personalizado"
            > </ Enumeración> </ restricción> </ simpleType> </ attribute> <atributo nombre = "local" tipo = "Booleano"
            > </ Attribute>

        </ SimpleType>

    </ Extension> </
    simpleContent> </ complexType>

    <complexType nombre = "LangString_f" >
    <SimpleContent>
        <extensión base = "cadena" > <Atributo nombre = "Lang" tipo = "cadena" /> </
        Extension> </ simpleContent> </ complexType>

    <complexType nombre = "LangStrings_f" > <Secuencia>

        <elemento nombre = "LangString_f" tipo = "Aas: langString_f" minOccurs = "0" maxOccurs = "ilimitado" > </ Element>
        </ Secuencia> </
        complexType>

```

```

<complexType nombre = "EmbeddedDataSpecification_f" > <Secuencia>

    <elemento nombre = "HasDataSpecification"
        tipo = "Aas: reference_f" maxOcurrencias = "1" minOccurs = "0" >
    </Element> <element nombre = "DataSpecificationContent"

        tipo = "Aas: dataSpecificationContent_f" maxOcurrencias = "1"
        minOccurs = "0" >
    </Element> </
secuencia> </
complexType>

<complexType nombre = "PrvalueType_f" > </ ComplexType>

<complexType nombre = "PropertyValue_f" > <SimpleContent>

    <extensión base = "cadena" > </ Extension>
</ SimpleContent> </
complexType> </ schema>

```

Nota: • designa de ajuste de línea para fines de diseño

### 3. AAS IEC61360 Tipo de datos

Para IEC 61360, una especificación de datos se pone a disposición, de forma individual:

```

<?xml versión = "1.0" codificación = "UTF-8" ?>
<Xsd: schema xmlns: xsd = "http://www.w3.org/2001/XMLSchema"
    targetNamespace = "http://www.admin-shell.io/IEC61360/1/0"
    xmlns: IEC61360 = "http://www.admin-shell.io/IEC61360/1/0"
    xmlns: AAS = "http://www.admin-shell.io/aas/1/0" xmlns: Q1 = "AAS" > <Xsd: import schemaLocation = "AAS.xsd"

    espacio de nombres = "http://www.admin-shell.io/aas/1/0" >
</ Xsd: import>

<Xsd: complexType nombre = "DataSpecificationIEC61360_f" > <Xsd: secuencia>

    <Xsd: elemento árbitro = "IEC61360: PreferredName" maxOcurrencias = "1"
        minOccurs = "1" >
    </ Xsd: element> <xsd: element árbitro = "IEC61360: shortName" maxOcurrencias = "1"

        minOccurs = "0" >
    </ Xsd: element> <xsd: element árbitro = "IEC61360: unidad" maxOcurrencias = "1"

        minOccurs = "0" >
    </ Xsd: element> <xsd: element árbitro = "IEC61360: UnitID" maxOcurrencias = "1"

        minOccurs = "0" >
    </ Xsd: element> <xsd: element árbitro = "IEC61360: ValueFormat" maxOcurrencias = "1"

        minOccurs = "0" >
    </ Xsd: element> <xsd: element árbitro = "IEC61360: sourceOfDefinition" maxOcurrencias = "1"

        minOccurs = "0" >
    </ Xsd: element> <xsd: element árbitro = "IEC61360: símbolo" maxOcurrencias = "1"

        minOccurs = "0" >
    </ Xsd: element> <xsd: element árbitro = "IEC61360: tipo de datos" maxOcurrencias = "1"

        minOccurs = "0" >
    </ Xsd: element> <xsd: element árbitro = "IEC61360: definición" maxOcurrencias = "1"

        minOccurs = "0" >
    </ Xsd: element> <xsd: element árbitro = "IEC61360: valueList" maxOcurrencias = "1"

        minOccurs = "0" >
    </ Xsd: element> <xsd: element árbitro = "IEC61360: código" maxOcurrencias = "1"

        minOccurs = "0" >
    </ Xsd: element> </ xsd:
secuencia> </ xsd: complexType>

<Xsd: complexType nombre = "Definition_f" > <Xsd: simpleContent>

    <Xsd: extension base = "Xsd: string" > <Xsd: atributo nombre = "Lang" tipo = "Xsd: string" />

```

```

</Xsd: extension> </xsd:
simpleContent> </xsd: complexType>

<Xsd: complexType nombre = "ValueList_f" > </Xsd: complexType>

<Xsd: complexType nombre = "Code_f" > </Xsd: complexType>

<Xsd: elemento nombre = "Nombre Preferido" tipo = "Aas: langStrings_f" > </Xsd: element>

<Xsd: elemento nombre = "nombre corto" tipo = "Xsd: string" > </Xsd: element>

<Xsd: elemento nombre = "unidad" tipo = "Xsd: string" > </Xsd: element>

<Xsd: elemento nombre = "UnitID" tipo = "Aas: reference_f" > </Xsd: element>

<Xsd: elemento nombre = "ValueFormal" tipo = "Xsd: string" > </Xsd: element>

<Xsd: elemento nombre = "SourceOfDefinition"
  tipo = "Aas: langStrings_f" >
</Xsd: element>

<Xsd: elemento nombre = "símbolo" tipo = "Xsd: string" > </Xsd: element>

<Xsd: elemento nombre = "tipo de datos" tipo = "Xsd: string" > </Xsd: element>

<Xsd: elemento nombre = "definición" tipo = "Xsd: string" > </Xsd: element>

<Xsd: elemento nombre = "tipo de valor" tipo = "IEC61360: valueList_f" > </Xsd: element>

<Xsd: elemento nombre = "código" tipo = "IEC61360: code_f" > </Xsd: element>

<Xsd: elemento nombre = "ValueList" tipo = "IEC61360: valueList_f" > </Xsd: element>
</Xsd: schema>

```

Nota: • designa de ajuste de línea para fines de diseño

## 4. Ejemplo XML

Para referencia cruzada, se da un ejemplo completo autónomo, que se refiere al ejemplo unificada en la cláusula 4.3.

```

<?xml versión = "1.0" codificación = "UTF-8" ?>
<Aas: aasenv xmlns: AAS = "http://www.admin-shell.io/aas/1/0"
  xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xmlns: IEC = "http://www.admin-shell.io/IEC61360/1/0"
  xsi: schemaLocation = "http://www.admin-shell.io/aas/1/0 AAS.xsd http://www.admin-shell.io/IEC61360/1/0 IEC61360.xsd" >

  <aas: assetAdministrationShells>
    <Aas: assetAdministrationShell>
      <Aas: identificación tipo de identificación = "URI" > www.admin-shell.io/aas-sample/1/0 </Aas: identificación> <aas: Administración>

      <versión> aas: 1 </Aas: Versión> <aas: revisión> 0 </Aas:
revisión> </aas: administración> <aas: assetRef>

      <aas: Teclas>
        <Aas: clave tipo = "Activo" local = "falso" tipo de identificación = "URI" > http://pk.festo.com/3s7plldrs35 </Aas: tecla> </aas: teclas> </aas: assetRef> <aas: submodelRefs>

      <Aas: submodelRef>
        <aas: Teclas>
          <Aas: clave tipo = "Submodelo" local = "cierto" •
            tipo de identificación = "URI" > "http://www.zvei.de/demo/submodel/12345679" </Aas: tecla> </aas: teclas> </aas: submodelRef> </
aas: submodelRefs> <aas: vistas>

      <aas: Ver>
        <Aas: idShort> SampleView </Aas: idShort> <aas: containedElements>

        <aas: containedElementRef>
          <aas: Teclas>
            <Aas: clave tipo = "Submodelo" local = "cierto"
              tipo de identificación = "URI" > "http://www.zvei.de/demo/submodel/12345679" </Aas: tecla> <aas: clave tipo = "Propiedad" local = "cierto" tipo de identificación =
              "idShort" > velocidad de rotación </Aas: tecla> </aas: teclas>

        </Aas: containedElementRef>

```

```

</ Aas: containedElements> </ aas: Vista> </
aas: vistas> <aas: conceptDictionaries>

<Aas: conceptDictionary>
  <Aas: idShort> SampleDic </ Aas: idShort> <aas:
conceptDescriptionRefs> <aas: conceptDescriptionRef> <aas: teclas>

    <Aas: clave tipo = "ConceptDescription" local = "cierto" tipo de identificación = "URI" > www.festo.com/dic/08111234 </ Aas: tecla> </ aas: teclas>

  </ Aas: conceptDescriptionRef> <aas:
conceptDescriptionRef>
  <aas: Teclas>
    <Aas: clave tipo = "ConceptDescription" local = "cierto" tipo de identificación = "IRDI" > 0173-1 # 02-BAA120 # 007 </ Aas: tecla> </ aas: teclas>

  </ Aas: conceptDescriptionRef> </ aas:
conceptDescriptionRefs> </ aas: conceptDictionary> </ aas:
conceptDictionaries> </ aas: assetAdministrationShell> </ aas:
assetAdministrationShells> <aas: activos>

<aas: activos>
  <Aas: idShort> 3s7plfdrs35 </ Aas: idShort> <aas: description>

    <Aas: LangString lang = "ES" > controlador de Festo </ Aas: LangString> </ aas: description> <aas: identificación tipo de identificación = "URI" > http://plk.festo.com/3s7plfdrs35
  </ Aas: identificación> <aas: tipo> Ejemplo </ Aas: tipo> </ aas: activo> </ aas: activos> <aas: submodelos>

<aas: submodelo>
  <Aas: identificación tipo de identificación = "URI" > http://www.zvei.de/demo/submodel/12345679 </ Aas: identificación> <aas: semanticId > <Aas: Teclas>

    <Aas: clave tipo de identificación = "URI" local = "falso"
      tipo = "Submodelo" > http://www.zvei.de/demo/submodelDefinitions/87654346 </ Aas: tecla> </ aas: teclas> </ aas: semanticId> <aas: tipo> Ejemplo </ aas:
      tipo> <aas: submodelElements>

<Aas: submodelElement>
  <aas: Vivienda>
    <Aas: idShort> velocidad de rotación </ Aas: idShort> <aas: category> VARIABLE
    </ Aas: categoría> <aas: semanticId>

    <aas: Teclas>
      <Aas: clave tipo de identificación = "URI" tipo = "ConceptDescription"
        local = "cierto" > www.festo.com/dic/08111234 </ Aas: tecla> </ aas: teclas> </ aas: semanticId>
      <aas: valueType> doble </ aas: valueType> </ aas: property> </ aas: submodelElement> <aas:
        submodelElement>

<aas: Vivienda>
  <Aas: idShort> NMAX </ Aas: idShort> <aas: category> PARÁMETRO </
  Aas: categoría> <aas: semanticId>

  <aas: Teclas>
    <Aas: clave tipo de identificación = "IRDI" tipo = "GlobalReference" local = "cierto" > 0173-1 # 02-BAA120 # 007 </ Aas: tecla> </ aas: teclas> </ aas: semanticId> <aas: valueType> doble
  </ Aas: valueType> <aas: valor> 2000 </ aas: valor> </ aas: property> </ aas: submodelElement> </ aas: submodelElements> </ aas: submodelo> </ aas: submodelos>

<aas: conceptDescriptions>
  <aas: conceptDescription>
    <Aas: identificación tipo de identificación = "URI" > www.festo.com/dic/08111234 </ Aas: identificación> <aas: embeddedDataSpecification>

    <aas: hasDataSpecification>
      <aas: Teclas>
        <Aas: clave tipo de identificación = "URI" local = "falso"
          tipo = "GlobalReference" > www.admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360 </ Aas: tecla>
        </ aas: Teclas>
      </ Aas: hasDataSpecification> <aas:
dataSpecificationContent>
      <Aas: dataSpecificationIEC61360>
        <IEC: PreferredName>
          <Aas: LangString lang = "DELAWARE" > Drehzahl </ Aas: LangString> <aas: LangString lang = "ES" > Velocidad de
          rotación </ Aas: LangString> </ IEC: PreferredName> <IEC: shortName> norte </ IEC: shortName> <IEC: UnitID>

        <aas: Teclas>

```

```

<Aas: clave local = "falso" tipo = "GlobalReference" tipo de identificación = "IRDI" > 0173-1 # 05-AAA650 # 002 </ Aas: tecla> </ aas: teclas> </ IEC: UnitID> <IEC: ValueFormat> NR1..5 </
IEC: ValueFormat> </ aas: dataSpecificationIEC61360> </ aas: dataSpecificationContent> </ aas: embeddedDataSpecification> </ aas: conceptDescription> <aas: conceptDescription>

<Aas: identificación tipo de identificación = "IRDI" > 0173-1 # 02-BAA120 # 007 </ Aas: identificación> <aas: embeddedDataSpecification>

<aas: hasDataSpecification>
  <aas: Teclas>
    <Aas: clave tipo de identificación = "URI" tipo = "GlobalReference" •
      local = "falso" > www.admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360 </ Aas: tecla> </ aas: teclas>

  </ Aas: hasDataSpecification>
  <Aas: dataSpecificationContent>
    <Aas: dataSpecificationIEC61360> <IEC: PreferredName>

      <Aas: LangString lang = "DELAWARE" > maximale Drehzahl </ Aas: LangString> <aas: LangString lang = "ES" > velocidad de
      rotación max </ Aas: LangString> </ IEC: PreferredName> <IEC: shortName> nMax </ IEC: shortName> <IEC: UnitID>
      <aas: teclas>

    <Aas: clave tipo = "GlobalReference" tipo de identificación = "IRDI" local = "falso" > 0173-1 # 05-AAA650 # 002 </ Aas: tecla> </ aas: teclas> </ IEC: UnitID> <IEC: ValueFormat> NR1..5 </
    IEC: ValueFormat> </ aas: dataSpecificationIEC61360> </ aas: dataSpecificationContent> </ aas: embeddedDataSpecification> </ aas: conceptDescription> </ aas: conceptDescriptions> </ aas:
    aasenv>

```

Nota: • designa de ajuste de línea para fines de diseño



# Anexo F. esquema JSON y ejemplos completos

## 1. JSON esquema para Shell Administración

El siguiente esquema utiliza JSON esquema <sup>28</sup> en el proyecto de la versión 04 para permitir la validación de archivos JSON.

Nota: este esquema es un modelo de núcleo; a partir de noviembre 2018, no figura el modelo de seguridad, puntos de vista y SubmodelElements seleccionados.

Tabla 17 esquema JSON

|   |   |
|---|---|
| <pre> {   "\$ Esquema": "http://json-schema.org/draft-04/schema#",   "titulo": "AssetAdministrationShellEnvironment",   "tipo": "objeto",   "additionalProperties": falso,   "necesario": [     "assetAdministrationShells",     "submodelos",     "bienes",     "conceptDescriptions"   ],   "propiedades": {     "assetAdministrationShells": {       "tipo": "formación",       "articulos": {         "\$ Ref": •       }     }   },   "# / Definiciones / AssetAdministrationShell"   },   "submodelos": {     "tipo": "formación",     "articulos": {       "\$ Ref": "# / Definiciones / submodelo"     }   },   "bienes": {     "tipo": "formación",     "articulos": {       "\$ Ref": "# / Definiciones / Activos"     }   },   "conceptDescriptions": {     "tipo": "formación",     "articulos": {       "\$ Ref": "# / Definiciones / ConceptDescription"     }   },   "Definiciones": {     "AssetAdministrationShell": {       "tipo": "objeto",       "additionalProperties": falso,       "propiedades": {         "identificación": {           "\$ Ref": "# / Definiciones / Identificación"         },         "administración": {           "\$ Ref": •         }       }     },     "# / Definiciones / AdministrativeInformation"     },     "IdShort": {       "tipo": "cuerda"     },     "categoria": {       "tipo": "cuerda"     },     "descripciones": {       "tipo": "formación",       "articulos": {         "\$ Ref": "# / Definiciones / Descripción"       }     },     "padre": {       "\$ Ref": "# / Definiciones / Reference"     },     "tipo de modelo": {       "\$ Ref": "# / Definiciones / ModelType"     }   },   "articulos": {     "\$ Ref": •   },   "# / Definiciones / EmbeddedDataSpecification"   },   "cualquiera de": [     {       "\$ Ref": "# / Definiciones / Propiedad"     },     {       "\$ Ref": "# / Definiciones / archivo"     },     {       "\$ Ref": "# / Definiciones / Blob"     },     {       "\$ Ref": "# / Definiciones / ReferenceElement"     },     {       "\$ Ref": •     },     "# / Definiciones / SubmodelElementCollection"     },     {       "\$ Ref": "# / Definiciones / RelationshipElement"     },     {       "\$ Ref": "# / Definiciones / Operación"     },     {       "\$ Ref": "# / Definiciones / OperationVariable"     }   ],   "Ver": {     "tipo": "objeto",     "additionalProperties": falso,     "propiedades": {       "SemanticId": {         "\$ Ref": "# / Definiciones / Reference"       },       "IdShort": {         "tipo": "cuerda"       },       "categoria": {         "tipo": "cuerda"       },       "descripciones": {         "tipo": "formación",         "articulos": {           "\$ Ref": "# / Definiciones / Descripción"         }       },       "padre": {         "\$ Ref": "# / Definiciones / Reference"       },       "tipo de modelo": {         "\$ Ref": "# / Definiciones / ModelType"       },       "containedElements": {         "tipo": "formación",         "articulos": {           "\$ Ref": "# / Definiciones / Reference"         }       }     },     "ConceptDictionary": { </pre> | <pre>     "articulos": {       "\$ Ref": •     },     "# / Definiciones / EmbeddedDataSpecification"     },     "cualquiera de": [       {         "\$ Ref": "# / Definiciones / Propiedad"       },       {         "\$ Ref": "# / Definiciones / archivo"       },       {         "\$ Ref": "# / Definiciones / Blob"       },       {         "\$ Ref": "# / Definiciones / ReferenceElement"       },       {         "\$ Ref": •       },       "# / Definiciones / SubmodelElementCollection"       },       {         "\$ Ref": "# / Definiciones / RelationshipElement"       },       {         "\$ Ref": "# / Definiciones / Operación"       },       {         "\$ Ref": "# / Definiciones / OperationVariable"       }     ],     "Ver": {       "tipo": "objeto",       "additionalProperties": falso,       "propiedades": {         "SemanticId": {           "\$ Ref": "# / Definiciones / Reference"         },         "IdShort": {           "tipo": "cuerda"         },         "categoria": {           "tipo": "cuerda"         },         "descripciones": {           "tipo": "formación",           "articulos": {             "\$ Ref": "# / Definiciones / Descripción"           }         },         "padre": {           "\$ Ref": "# / Definiciones / Reference"         },         "tipo de modelo": {           "\$ Ref": "# / Definiciones / ModelType"         },         "containedElements": {           "tipo": "formación",           "articulos": {             "\$ Ref": "# / Definiciones / Reference"           }         }       },       "ConceptDictionary": { </pre> |
|---|---|

<sup>28</sup> ver: <http://json-schema.org/>

|  |   |
|--|---|
| <pre> "embeddedDataSpecifications" : {   "tipo" : "formación",   "articulos" : {     "\$ Ref" : •   } }  "/ Definiciones / EmbeddedDataSpecification"  "derivado de" : {   "\$ Ref" : "# / Definiciones / Reference" }  "activo" : {   "\$ Ref" : "# / Definiciones / Reference" }  "submodelos" : {   "tipo" : "formación",   "articulos" : {     "\$ Ref" : "# / Definiciones / Reference"   } }  "puntos de vista" : {   "tipo" : "formación",   "articulos" : {     "\$ Ref" : "# / Definiciones / Vista"   } }  "conceptDictionaries" : {   "tipo" : "formación",   "articulos" : {     "\$ Ref" : "# / Definiciones / ConceptDictionary"   } }  "Identifier" : {   "tipo" : "objeto",   "additionalProperties" : falso,   "propiedades" : {     "carté de identidad" : {       "tipo" : "cuerda"     },     "tipo de identificación" : {       "uno de" : [         {           "tipo" : "nulo"         }       ]     }   }   "\$ Ref" : "# / Definiciones / KeyType" }  "KeyType" : {   "tipo" : "cuerda",   "descripción" : "",   "X-enumNames" : {     "Personalizado",     "URI",     "IRDI",     "IdShort"   } }  "Enumeración" : {   "Personalizado",   "URI",   "IRDI",   "IdShort" }  "Información administrativa" : {   "tipo" : "objeto",   "additionalProperties" : falso,   "propiedades" : {     "versión" : {       "tipo" : "cuerda"     },     "revisión" : {       "tipo" : "cuerda"     }   } }  "Descripción" : {   "tipo" : "objeto",   "additionalProperties" : falso,   "propiedades" : {     "idioma" : {       "tipo" : "cuerda"     },     "texto" : {       "tipo" : "cuerda"     }   } } </pre> | <pre> "tipo" : "objeto", "additionalProperties" : falso, "propiedades" : {   "IdShort" : {     "tipo" : "cuerda"   },   "categoría" : {     "tipo" : "cuerda"   },   "descripciones" : {     "tipo" : "formación",     "articulos" : {       "\$ Ref" : "# / Definiciones / Descripción"     }   },   "padre" : {     "\$ Ref" : "# / Definiciones / Reference"   },   "conceptDescriptions" : {     "tipo" : "formación",     "articulos" : {       "\$ Ref" : "# / Definiciones / Reference"     }   } }  "ConceptDescription" : {   "tipo" : "objeto",   "additionalProperties" : falso,   "propiedades" : {     "identificación" : {       "\$ Ref" : "# / Definiciones / Identificación"     },     "administración" : {       "\$ Ref" : •     }   } }  "/ Definiciones / AdministrativeInformation"  "IdShort" : {   "tipo" : "cuerda" }  "categoría" : {   "tipo" : "cuerda" }  "descripciones" : {   "tipo" : "formación",   "articulos" : {     "\$ Ref" : "# / Definiciones / Descripción"   } }  "padre" : {   "\$ Ref" : "# / Definiciones / Reference" }  "embeddedDataSpecifications" : {   "tipo" : "formación",   "articulos" : {     "\$ Ref" : •   } }  "/ Definiciones / EmbeddedDataSpecification"  "tipo de modelo" : {   "\$ Ref" : "# / Definiciones / ModelType" }  "IsCaseOf" : {   "tipo" : "formación",   "articulos" : {     "\$ Ref" : "# / Definiciones / Reference"   } }  "Propiedad" : {   "tipo" : "objeto",   "additionalProperties" : falso,   "propiedades" : {     "valor" : {},     "tipo de valor" : {       "\$ Ref" : "# / Definiciones / Tipo de datos"     }   },   "tipo de modelo" : {     "\$ Ref" : "# / Definiciones / ModelType"   },   "IdShort" : {     "tipo" : "cuerda"   },   "SemanticId" : {     "\$ Ref" : "# / Definiciones / Reference"   },   "limitaciones" : {     "tipo" : "formación",     "articulos" : { </pre> |
|--|---|

```

    }},

    "Referencia" : {
        "tipo" : "objeto",
        "additionalProperties" : falso,
        "propiedades" : {
            "llaves" : {
                "tipo" : "formación",
                "articulos" : {
                    "$ Ref" : "# / Definiciones / Clave"
                }
            }
        }
    }

    "Llave" : {
        "tipo" : "objeto",
        "additionalProperties" : falso,
        "propiedades" : {
            "tipo" : {
                "uno de" : [
                    {
                        "tipo" : "nulo"
                    }, {
                        "$ Ref" : "# / definiciones / KeyElements"
                    }
                ]
            },

            "tipo de identificación" : {
                "uno de" : [
                    {
                        "tipo" : "nulo"
                    }, {
                        "$ Ref" : "# / Definiciones / KeyType"
                    }
                ]
            },

            "valor" : {
                "tipo" : "cuerda"
            },
            "local" : {
                "tipo" : [
                    "Booleano",
                    "nulo"
                ]
            },

            "índice" : {
                "tipo" : [
                    "entero",
                    "nulo"
                ],
                "formato" : "int32"
            }
        }
    }

    "Elementos clave" : {
        "tipo" : "cuerda",
        "descripción" : "",
        "X-enunNames" : [
            "GlobalReference",
            "ConceptDictionary",
            "AccessPermissionRule",
            "DataElement",
            "Ver",
            "Propiedad",
            "SubmodelElement",
            "Expediente",
            "Gota",
            "ReferenceElement",
            "SubmodelElementCollection",
            "RelationshipElement",
            "Evento",
            "Operación",
            "OperationParameter",
            "AssetAdministrationShell",
            "Submodelo",
            "ConceptIDescription",
            "Activo"
        ],
        "Enumeración" : [
            "GlobalReference",
            "ConceptDictionary",
            "AccessPermissionRule",
            "DataElement",
            "Ver",
            "Propiedad",
            "SubmodelElement",
            "Expediente",
            "Gota",

```

```

        "$ Ref" : "# / Definiciones / Restricción"
    }},

    "categoría" : {
        "tipo" : "cuerda"
    },
    "descripciones" : {
        "tipo" : "formación",
        "articulos" : {
            "$ Ref" : "# / Definiciones / Descripción"
        }
    },

    "padre" : {
        "$ Ref" : "# / Definiciones / Reference"
    },
    "tipo" : {
        "uno de" : [
            {
                "tipo" : "nulo"
            }, {
                "$ Ref" : "# / Definiciones / Kind"
            }
        ]
    },

    "embeddedDataSpecifications" : {
        "tipo" : "formación",
        "articulos" : {
            "$ Ref" : •
        }
    },
    "# / Definiciones / EmbeddedDataSpecification"
    },

    "ValueId" : {
        "$ Ref" : "# / Definiciones / Reference"
    }
    },

    "Expediente" : {
        "tipo" : "objeto",
        "additionalProperties" : falso,
        "propiedades" : {
            "valor" : {
                "tipo" : "cuerda"
            },
            "tipo de valor" : {
                "$ Ref" : "# / Definiciones / Tipo de datos"
            },
            "tipo de modelo" : {
                "$ Ref" : "# / Definiciones / ModelType"
            },
            "IdShort" : {
                "tipo" : "cuerda"
            },
            "SemanticId" : {
                "$ Ref" : "# / Definiciones / Reference"
            },
            "limitaciones" : {
                "tipo" : "formación",
                "articulos" : {
                    "$ Ref" : "# / Definiciones / Restricción"
                }
            },
            "categoría" : {
                "tipo" : "cuerda"
            },
            "descripciones" : {
                "tipo" : "formación",
                "articulos" : {
                    "$ Ref" : "# / Definiciones / Descripción"
                }
            },

            "padre" : {
                "$ Ref" : "# / Definiciones / Reference"
            },
            "tipo" : {
                "uno de" : [
                    {
                        "tipo" : "nulo"
                    }, {
                        "$ Ref" : "# / Definiciones / Kind"
                    }
                ]
            },

            "embeddedDataSpecifications" : {
                "tipo" : "formación",
                "articulos" : {
                    "$ Ref" :
                }
            },
            "# / Definiciones / EmbeddedDataSpecification"
        }
    }

```

```

"ReferenceElement" ,
"SubmodelElementCollection" ,
"RelationshipElement" ,
"Evento" ,
"Operación" ,
"OperationParameter" ,
"AssetAdministrationShell" ,
"Submodelo" ,
"ConceptIDescription" ,
"Activo"
]],

"Tipo de modelo" : {
  "tipo" : "objeto" ,
  "additionalProperties" : falso ,
  "propiedades" : {
    "nombre" : {
      "tipo" : "cuerda"
    }
  }
}],

"EmbeddedDataSpecification" : {
  "tipo" : "objeto" ,
  "additionalProperties" : falso ,
  "propiedades" : {
    "HasDataSpecification" : {
      "$ Ref" : "# / Definiciones / Reference"
    }
  },
  "DataSpecificationContent" : {
    "$ Ref" : •
  }
}
"# / Definiciones / DataSpecificationContent"
]],

"DataSpecificationContent" : {
  "tipo" : "objeto"
},
"Activo" : {
  "additionalProperties" : falso ,
  "propiedades" : {
    "identificación" : {
      "$ Ref" : "# / Definiciones / Identificación"
    }
  },
  "administración" : {
    "$ Ref" : •
  }
}
"# / Definiciones / AdministrativeInformation"
]],

"IdShort" : {
  "tipo" : "cuerda"
},
"categoria" : {
  "tipo" : "cuerda"
},
"descripciones" : {
  "tipo" : "formación" ,
  "articulos" : {
    "$ Ref" : "# / Definiciones / Descripción"
  }
}],

"padre" : {
  "$ Ref" : "# / Definiciones / Reference"
},
"tipo" : {
  "uno de" : [
    {
      "tipo" : "nulo"
    }
  ], {
    "$ Ref" : "# / Definiciones / Kind"
  }
}],

"SemantId" : {
  "$ Ref" : "# / Definiciones / Reference"
},
"tipo de modelo" : {
  "$ Ref" : "# / Definiciones / ModelType"
},
"embeddedDataSpecifications" : {
  "tipo" : "formación" ,
  "articulos" : {
    "$ Ref" : •
  }
}
"# / Definiciones / EmbeddedDataSpecification"
]],

"AssetIdentificationModel" : {
  "$ Ref" : "# / Definiciones / Reference"
}],

"tipo MIME" : {
  "tipo" : "cuerda"
}],

"Gota" : {
  "tipo" : "objeto" ,
  "additionalProperties" : falso ,
  "propiedades" : {
    "valor" : {
      "tipo" : "cuerda" ,
      "formato" : "byte"
    }
  },
  "tipo de valor" : {
    "$ Ref" : "# / Definiciones / Tipo de datos"
  },
  "tipo de modelo" : {
    "$ Ref" : "# / Definiciones / ModelType"
  },
  "IdShort" : {
    "tipo" : "cuerda"
  },
  "SemantId" : {
    "$ Ref" : "# / Definiciones / Reference"
  },
  "limitaciones" : {
    "tipo" : "formación" ,
    "articulos" : {
      "$ Ref" : "# / Definiciones / Restricción"
    }
  }
}],

"categoria" : {
  "tipo" : "cuerda"
},
"descripciones" : {
  "tipo" : "formación" ,
  "articulos" : {
    "$ Ref" : "# / Definiciones / Descripción"
  }
}],

"padre" : {
  "$ Ref" : "# / Definiciones / Reference"
},
"tipo" : {
  "uno de" : [
    {
      "tipo" : "nulo"
    }
  ], {
    "$ Ref" : "# / Definiciones / Kind"
  }
}],

"embeddedDataSpecifications" : {
  "tipo" : "formación" ,
  "articulos" : {
    "$ Ref" : •
  }
}
"# / Definiciones / EmbeddedDataSpecification"
]],

"tipo MIME" : {
  "tipo" : "cuerda"
}],

"ReferenceElement" : {
  "tipo" : "objeto" ,
  "additionalProperties" : falso ,
  "propiedades" : {
    "valor" : {
      "$ Ref" : "# / Definiciones / Reference"
    }
  },
  "tipo de valor" : {
    "$ Ref" : "# / Definiciones / Tipo de datos"
  },
  "tipo de modelo" : {
    "$ Ref" : "# / Definiciones / ModelType"
  },
  "IdShort" : {
    "tipo" : "cuerda"
  },
  "SemantId" : {
    "$ Ref" : "# / Definiciones / Reference"
  },
  "limitaciones" : {
    "tipo" : "formación" ,
    "articulos" : {
      "$ Ref" : "# / Definiciones / Restricción"
    }
  }
}

```

```

"Tipo": {
  "tipo": "cuerda",
  "descripción": "",
  "X-enumNames": [
    "Tipo",
    "Ejemplo"
  ],
  "Enumeración": [
    "Tipo",
    "Ejemplo"
  ]
},

"Submodelo": {
  "additionalProperties": falso,
  "propiedades": {
    "identificación": {
      "$ Ref": "# / Definiciones / Identificación"
    },
    "administración": {
      "$ Ref": •
    }
  }
}

"# / Definiciones / AdministrativeInformation"
},
"IdShort": {
  "tipo": "cuerda"
},
},
"categoria": {
  "tipo": "cuerda"
},
},
"descripciones": {
  "tipo": "formación",
  "articulos": {
    "$ Ref": "# / Definiciones / Descripción"
  }
},
},
"padre": {
  "$ Ref": "# / Definiciones / Reference"
},
},
"tipo": {
  "uno de": [
    {
      "tipo": "nulo"
    },
    {
      "tipo": "nulo"
    }
  ]
},
},
"$ Ref": "# / Definiciones / Kind"
}],
},
},

"Semantid": {
  "$ Ref": "# / Definiciones / Reference"
},
},
"tipo de modelo": {
  "$ Ref": "# / Definiciones / ModelType"
},
},
"embeddedDataSpecifications": {
  "tipo": "formación",
  "articulos": {
    "$ Ref": •
  }
},
},
"# / Definiciones / EmbeddedDataSpecification"
}],
},
},

"submodelElements": {
  "tipo": "formación",
  "articulos": {
    "$ Ref": "# / Definiciones / SubmodelElement"
  }
},
},
},

"Restricción": {
  "tipo": "objeto",
  "additionalProperties": falso,
  "propiedades": {
    "tipo de modelo": {
      "$ Ref": "# / Definiciones / ModelType"
    }
  },
},
},
"cualequiera de": [
  {
    "$ Ref": "# / Definiciones / Fórmula"
  },
  {
    "$ Ref": "# / Definiciones / Calificador"
  }
],
},
},

"Tipo de datos": {
  "tipo": "objeto",
  "additionalProperties": falso,
  "propiedades": {
    "DataObjectType": {

```

```

},
},
"categoria": {
  "tipo": "cuerda"
},
},
"descripciones": {
  "tipo": "formación",
  "articulos": {
    "$ Ref": "# / Definiciones / Descripción"
  }
},
},
"padre": {
  "$ Ref": "# / Definiciones / Reference"
},
},
"tipo": {
  "uno de": [
    {
      "tipo": "nulo"
    },
    {
      "tipo": "nulo"
    }
  ]
},
},
"$ Ref": "# / Definiciones / Kind"
}],
},
},

"embeddedDataSpecifications": {
  "tipo": "formación",
  "articulos": {
    "$ Ref": •
  }
},
},
"# / Definiciones / EmbeddedDataSpecification"
}],
},
},

"SubmodelElementCollection": {
  "tipo": "objeto",
  "additionalProperties": falso,
  "propiedades": {
    "valor": {
      "tipo": "formación",
      "articulos": {
        "$ Ref": "# / Definiciones / SubmodelElement"
      }
    },
    "tipo de valor": {
      "$ Ref": "# / Definiciones / Tipo de datos"
    },
    "tipo de modelo": {
      "$ Ref": "# / Definiciones / ModelType"
    },
    "IdShort": {
      "tipo": "cuerda"
    },
    "Semantid": {
      "$ Ref": "# / Definiciones / Reference"
    },
    "limitaciones": {
      "tipo": "formación",
      "articulos": {
        "$ Ref": "# / Definiciones / Restricción"
      }
    },
    "categoria": {
      "tipo": "cuerda"
    },
    "descripciones": {
      "tipo": "formación",
      "articulos": {
        "$ Ref": "# / Definiciones / Descripción"
      }
    },
    "padre": {
      "$ Ref": "# / Definiciones / Reference"
    },
    "tipo": {
      "uno de": [
        {
          "tipo": "nulo"
        },
        {
          "tipo": "nulo"
        }
      ]
    },
    "$ Ref": "# / Definiciones / Kind"
  }
},
},
},

"embeddedDataSpecifications": {
  "tipo": "formación",
  "articulos": {
    "$ Ref": •
  }
},
},
"# / Definiciones / EmbeddedDataSpecification"
}],
},
},

"AllowDuplicates": {

```

|   |  |
|---|--|
| <pre> "\$ Ref" : "# / Definiciones / DataObjectType"     }   }    "DataObjectType" : {     "tipo" : "objeto",     "additionalProperties" : falso ,     "propiedades" : {       "nombre" : {         "tipo" : "cuerda"       }     }   }    "Operación" : {     "tipo" : "objeto",     "additionalProperties" : falso ,     "propiedades" : {       "Semantictid" : {         "\$ Ref" : "# / Definiciones / Reference"       }     }     "limitaciones" : {       "tipo" : "formación",       "articulos" : {         "\$ Ref" : "# / Definiciones / Restricción"       }     }      "IdShort" : {       "tipo" : "cuerda"     }     "categoria" : {       "tipo" : "cuerda"     }     "descripciones" : {       "tipo" : "formación",       "articulos" : {         "\$ Ref" : "# / Definiciones / Descripción"       }     }      "padre" : {       "\$ Ref" : "# / Definiciones / Reference"     }     "tipo" : {       "uno de" : [         {           "tipo" : "nulo"         }       ]     }      "\$ Ref" : "# / Definiciones / Kind"   }    "tipo de modelo" : {     "\$ Ref" : "# / Definiciones / ModelType"   }    "embeddedDataSpecifications" : {     "tipo" : "formación",     "articulos" : {       "\$ Ref" : •     }   } "# / Definiciones / EmbeddedDataSpecification"   }    "en" : {     "tipo" : "formación",     "articulos" : {       "\$ Ref" : "# / Definiciones / OperationVariable"     }   }    "afuera" : {     "tipo" : "formación",     "articulos" : {       "\$ Ref" : "# / Definiciones / OperationVariable"     }   }    "OperationVariable" : {     "tipo" : "objeto",     "additionalProperties" : falso ,     "propiedades" : {       "Semantictid" : {         "\$ Ref" : "# / Definiciones / Reference"       }     }     "limitaciones" : {       "tipo" : "formación",       "articulos" : {         "\$ Ref" : "# / Definiciones / Restricción"       }     }      "IdShort" : {       "tipo" : "cuerda"     } </pre> | <pre> "tipo" : "Booleano"   }   "ordenado" : {     "tipo" : "Booleano"   } }  "RelationshipElement" : {   "tipo" : "objeto",   "additionalProperties" : falso ,   "propiedades" : {     "IdShort" : {       "tipo" : "cuerda"     }   }   "Semantictid" : {     "\$ Ref" : "# / Definiciones / Reference"   }   "limitaciones" : {     "tipo" : "formación",     "articulos" : {       "\$ Ref" : "# / Definiciones / Restricción"     }   }    "categoria" : {     "tipo" : "cuerda"   }   "descripciones" : {     "tipo" : "formación",     "articulos" : {       "\$ Ref" : "# / Definiciones / Descripción"     }   }    "padre" : {     "\$ Ref" : "# / Definiciones / Reference"   }   "tipo" : {     "uno de" : [       {         "tipo" : "nulo"       }     ]   }    "\$ Ref" : "# / Definiciones / Kind" }  "tipo de modelo" : {   "\$ Ref" : "# / Definiciones / ModelType" }  "embeddedDataSpecifications" : {   "tipo" : "formación",   "articulos" : {     "\$ Ref" : •   } "# / Definiciones / EmbeddedDataSpecification"   }    "primero" : {     "\$ Ref" : "# / Definiciones / Reference"   }   "segundo" : {     "\$ Ref" : "# / Definiciones / Reference"   } }  "Índice" : {   "tipo" : "objeto",   "additionalProperties" : falso ,   "propiedades" : {     "tipo de modelo" : {       "\$ Ref" : "# / Definiciones / ModelType"     }   }   "Semantictid" : {     "\$ Ref" : "# / Definiciones / Reference"   }   "QualifierType" : {     "tipo" : "cuerda"   }   "QualifierValue" : {},   "QualifierValueId" : {     "\$ Ref" : "# / Definiciones / Reference"   } }  "Fórmula" : {   "tipo" : "objeto",   "additionalProperties" : falso ,   "propiedades" : {     "tipo de modelo" : {       "\$ Ref" : "# / Definiciones / ModelType"     }   }   "depende de" : { </pre> |
|---|--|

```

    },
    "categoria" : {
        "tipo" : "cuerda"
    },
    "descripciones" : {
        "tipo" : "formación",
        "articulos" : {
            "$ Ref" : "# / Definiciones / Descripción"
        }
    },

    "padre" : {
        "$ Ref" : "# / Definiciones / Reference"
    },
    "tipo" : {
        "uno de" : [
            {
                "tipo" : "nulo"
            }, {

                "$ Ref" : "# / Definiciones / Kind"
            }
        ]
    },

    "tipo de modelo" : {
        "$ Ref" : "# / Definiciones / ModelType"
    },
    "embeddedDataSpecifications" : {
        "tipo" : "formación",
        "articulos" : {
            "$ Ref" : •
"# / Definiciones / EmbeddedDataSpecification"
        }
    },

    "índice" : {
        "tipo" : [
            "entero",
            "nulo"
        ],
        "formato" : "int32"
    },
    "tipo de datos" : {
        "$ Ref" : "# / Definiciones / Tipo de datos"
    }
}],

"SubmodelElement" : {
    "tipo" : "objeto",
    "propiedades" : {
        "Semantidcl" : {
            "$ Ref" : "# / Definiciones / Reference"
        }
    },
    "limitaciones" : {
        "tipo" : "formación",
        "articulos" : {
            "$ Ref" : "# / Definiciones / Restricción"
        }
    },

    "IdShort" : {
        "tipo" : "cuerda"
    },
    "categoria" : {
        "tipo" : "cuerda"
    },
    "descripciones" : {
        "tipo" : "formación",
        "articulos" : {
            "$ Ref" : "# / Definiciones / Descripción"
        }
    },

    "padre" : {
        "$ Ref" : "# / Definiciones / Reference"
    },
    "tipo" : {
        "uno de" : [
            {
                "tipo" : "nulo"
            }, {

                "$ Ref" : "# / Definiciones / Kind"
            }
        ]
    },

    "tipo de modelo" : {
        "$ Ref" : "# / Definiciones / ModelType"
    },
    "embeddedDataSpecifications" : {
        "tipo" : "formación",

```

```
"tipo": "formación",
"articulos": {
    "$ Ref": "# / Definiciones / Reference"
}
```

## 2. Ejemplo JSON

Para referencia cruzada, se da un ejemplo completo autónomo, que se refiere al ejemplo unificada en la cláusula 4.3.

Tabla 18 JSON ejemplo completo

|   |   |
|---|---|
| <pre>{   "assetAdministrationShells": [     {       "identificación": {         "carné de identidad": "Www.admin-shell.io/aas-sample/1.0",         "tipo de identificación": "URI"       },       "activo": {         "llaves": [           {             "tipo": "Activo",             "tipo de identificación": "URI",             "valor": *           }         ],         "Http://pk.festo.com/3S7PLFDRS35",         "local": cierto,         "índice": 0       }     }   ],    "submodelos": [     {       "llaves": [         {           "tipo": "Submodelo",           "tipo de identificación": "URI",           "valor": *         }       ],       "Http://www.zvei.de/demo/submodel/12345679",       "local": cierto,       "índice": 0     }   ] },    "puntos de vista": [     {       "IdShort": "SampleView",       "containedElements": [         {           "llaves": [             {               "tipo": "Propiedad",               "tipo de identificación": "IdShort",               "valor": "velocidad de rotación",               "local": cierto,               "índice": 0             }           ]         }       ]     }   ],    "tipo de modelo": {     "nombre": "Ver"   } },    "administración": {     "versión": "1",     "revisión": "0"   },   "tipo de modelo": {     "nombre": "AssetAdministrationShell"   },   "conceptDictionaries": [     {       "conceptDescriptions": [         {           "llaves": [             {               "tipo": "ConceptDescription",               "tipo de identificación": "URI",               "valor": *             }           ],           "local": cierto,           "índice": 0         }       ]     }   ],   "Www.festo.com/dic/08111234",    "llaves": [     {       "tipo": "ConceptDescription",</pre> | <pre>"bienes": [   {     "IdShort": "3S7PLFDRS35",     "identificación": {       "carné de identidad": "Http://pk.festo.com/3S7PLFDRS35",       "tipo de identificación": "URI"     },     "tipo": "Ejemplo",     "descripciones": [       {         "idioma": "ES",         "texto": "Controller Festo"       }     ],     "tipo de modelo": {       "nombre": "Activo"     }   } ],    "conceptDescriptions": [     {       "embeddedDataSpecifications": [         {           "HasDataSpecification": {             "llaves": [               {                 "tipo": "GlobalReference",                 "tipo de identificación": "URI",                 "valor": "www.admin- * shell.io/DataSpecificationTemplates/DataSpecificationIEC613 60",                  "local": falso,                 "índice": 0               }             ]           },            "DataSpecificationContent": {             "Nombre Preferido": {               "idioma": "ES",               "texto": "Velocidad de rotación"             },             "nombre corto": "NORTE",             "UnitID": {               "llaves": [                 {                   "tipo": "GlobalReference",                   "tipo de identificación": "IRDI",                   "valor": "0173-1 # 05-AA650 # 002",                   "local": falso,                   "índice": 0                 }               ]             },              "ValueFormat": "NR1..5"           }         }       ]     },      "identificación": {       "carné de identidad": "Www.festo.com/dic/08111234",       "tipo de identificación": "URI"     },     "tipo de modelo": {       "nombre": "ConceptDescription"     }   } ],    "embeddedDataSpecifications": [     {       "HasDataSpecification": {         "llaves": [           {             "tipo": "GlobalReference",             "tipo de identificación": "URI",             "valor": "www.admin- * shell.io/DataSpecificationTemplates/DataSpecificationIEC613 60",              "local": falso,             "índice": 0           }         ]       }     }   ] }</pre> |
|---|---|



```

        "tipo de identificación" : "IRDI",
        "valor" : "0173-1 # 02-BAA120 # 007",
        "local" : cierto ,
        "índice" : 0
    }
}
}
}
}

"submodelos" : [
{
    "identificación" : {
        "carné de identidad" :
"http://www.zvei.de/demo/submodel/12345679",
        "tipo de identificación" : "URI"
    },
    "tipo" : "Ejemplo",
    "SemanticId" : {
        "llaves" : [
            {
                "tipo" : "GlobalReference",
                "tipo de identificación" : "URI",
                "valor" : •
            }
        ],
        "local" : falso ,
        "índice" : 0
    }
}],

"submodelElements" : [
{
    "IdShort" : "velocidad de rotación",
    "tipo de modelo" : {
        "nombre" : "Propiedad"
    },
    "tipo de valor" : {
        "DataObjectType" : {
            "nombre" : "doble"
        }
    },
    "SemanticId" : {
        "llaves" : [
            {
                "tipo" : "ConceptDescription",
                "tipo de identificación" : "URI",
                "valor" : •
            }
        ],
        "local" : cierto ,
        "índice" : 0
    }
}],

"categoria" : "VARIABLE"
}, {

    "IdShort" : "NMAX",
    "tipo de modelo" : {
        "nombre" : "Propiedad"
    },
    "tipo de valor" : {
        "DataObjectType" : {
            "nombre" : "doble"
        }
    },
    "SemanticId" : {
        "llaves" : [
            {
                "tipo" : "ConceptDescription",
                "tipo de identificación" : "IRDI",
                "valor" : "0173-1 # 02-BAA120 # 007",
                "local" : cierto ,
                "índice" : 0
            }
        ]
    },

    "categoria" : "PARÁMETRO"
}],

"tipo de modelo" : {
    "nombre" : "Submodelo"
}],
    
```

**Nota:** por encima de contenido está envuelto en varias columnas; • designa de ajuste de línea para fines de diseño

## Anexo G. Bibliografía

- [1] Recomendaciones para la implementación de la iniciativa estratégica INDUSTRIE 4.0; acatech; Abril 2013; Plattform Industrie 4.0 oficina administrativa;  
<https://www.acatech.de/Publikation/recommendations-for-implementing-the-strategic-initiativeindustrie-4-0-final-report-of-the-industrie-4-0-work>

---

- [2] Estrategia de ejecución Industrie 4.0: Informe sobre los resultados de la Plataforma 4.0 Industrie; BITKOM eV, VDMA eV, ZVEI eV; De abril de 2015;  
<https://www.bitkom.org/noindex/Publikationen/2016/Sonstiges/Implementation-StrategyIndustrie-40/2016-01-Implementation-Strategy-Industrie>

---

- [3] DIN SPEC 91345: 2016-04, Referenzarchitekturmodell Industrie 4,0 (RAMI4.0) (DIN SPEC 91345: 2016-04, la arquitectura de referencia de modelo Industrie 4,0 (RAMI4.0)),  
<https://www.beuth.de/en/technical-rule/din-spec-91345-en/250940128>

---

- [4] Resultado de papel "Estructura de la Administración de Shell, la continuación del desarrollo del modelo de referencia para el componente Industrie 4.0"; Plattform Industrie 4.0; abril de 2016;  
<https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/struktur-derverwaltungsschale.html>;  
<http://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/structure-of-the-administration-shell.html>

---

- [5] LEITFADEN "Welche Kriterien müssen Industrie-4,0-Produkte erfüllen?" (Directrices "¿Qué criterios deben Industrie 4.0 productos cumplir?); ZVEI eV; De noviembre de de 2016; <https://www.zvei.org/pressemedien/publikationen/welche-kriterien-muessen-industrie-4-0>;  
; Versión Inglés planeado

---

- [6] Libro Blanco "Beispiele zur Verwaltungsschale der Industrie 4,0-Komponente - Basisteil" (Ejemplos para la Shell Administración de Industrie 4.0 Componentes - parte básica); ZVEI eV; De noviembre de de 2016; <https://www.zvei.org/presse-medien/publikationen>;  
; Versión Inglés planeado

---

- [7] Industrie 4.0 Documento de trabajo "Aspectos de la hoja de ruta de investigación en escenarios de aplicación", Plattform Industrie 4.0, Abril de 2016; <http://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/anwendungsszenarien-aufforschungsroadmap.html>;  
<http://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/aspects-of-the-research-roadmap.html>

---

- [8] Industrie 4,0 documento de trabajo "Fortschreibung der Anwendungsszenarien" (Continuación de los escenarios de aplicación); Plattform Industrie 4,0; De octubre de de 2016; <https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/fortschreibung-a>  
<http://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/aspects-of-the-researchroadmap.html>

---

- [9] Descripción técnica general "Sichere Identitäten" (Secure identidades); Berlina; Plattform Industrie 4.0; 2016;  
<https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/sichere-identitaeten.html>;  
<http://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/security-rami40-en.html>

---

- [10] DKE Deutsche Kommission Elektrotechnik, Elektronik Informationstechnik im DIN VDE und: Die Deutsche Normungs-Roadmap Industrie 4.0 (La estandarización hoja de ruta alemán Industrie 4.0); Version 2.0; 2015; <http://www.din.de/de/forschung-und-innovation/industrie4-0/roadmapindustrie40-62178>

---

- [10] Documento de debate "Weiterentwicklung des Interaktionsmodells für Industrie 4,0-Komponenten" (Un mayor desarrollo de modelo de interacción para Industrie 4.0 componentes); Plattform Industrie 4,0; De noviembre de de 2016; <https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/interaktionsmodell-i40-komponenten-itgipfel.html>
- [11] Definición de términos relacionados con Industrie 4.0; Sitio web; Fraunhofer IOSB y VDI GMA Fachausschuss 7,21; <http://i40.iosb.fraunhofer.de/FA>; consultado el 02/12/2017
- [12] Industrie 4.0 documento de trabajo "Beziehungen zwischen I4.0-Componentes - Verbundkomponenten Produktion und intelligente" (Relaciones entre los componentes I4.0 - Componentes compuestos y la producción inteligente); Berlina; Plattform Industrie 4,0; De junio de 2017. <http://www.plattformi40.de/I40/Redaktion/DE/Downloads/Publikation/beziehungen-%20i40-komponenten.html>
- [13] Industrie 4.0 documento de trabajo "Industrie 4.0 Plug-and-Produce para las fábricas Adaptable"; Berlina; Plattform Industrie 4,0; De junio de 2017. <http://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/Industrie-40-%20Plug-and-Produce.html>
- [14] Industrie 4.0 documento de trabajo "Sicherheit der Verwaltungsschale" (Seguridad de la Concha de Administración); Berlina; Plattform Industrie 4,0; De abril de 2017. <http://www.plattformi40.de/I40/Redaktion/DE/Downloads/Publikation/security-der-verwaltungsschale.html>
- [15] DIN 92000 SPEC, "Estados Propiedad Valor", en curso
- [dieciséis] Verwaltungsschale Konkret (Específicos de Shell Administración), GMA Fachausschuss 7,20, en curso
- [17] Semántica y la interacción de los componentes I4.0, título de trabajo "Sprache für I4.0-Komponenten" (Lenguaje para I4.0 componentes), GMA grupo tecnología de 7,20, en curso
- [18] La estructura de la Administración de Shell: PERSPECTIVAS TRILATERALES de Francia, Italia y Alemania. De marzo de de 2018; [https://www.plattformi40.de/I40/Redaktion/DE/Downloads/Publikation/hm-2018-trilateralecoop.pdf?\\_\\_blob=publicationFile&v=9](https://www.plattformi40.de/I40/Redaktion/DE/Downloads/Publikation/hm-2018-trilateralecoop.pdf?__blob=publicationFile&v=9)
- [19] Zugriffssteuerung für Industrie 4,0-Komponenten zur Anwendung von Herstellern, Betreibern und Integratoren (control de acceso para Industrie 4.0 Componentes), Plattform Industrie 4,0 UAG Rollen und Rechte, en curso
- [20] ISO 29002-10 sistemas de automatización industrial e integración - Intercambio de datos característicos - Parte 10: formato de intercambio de datos característicos. Especificación Técnica ISO / TS 29002-10: 2009 (E)
- [21] B. Otto; S. Lohmann et al Arquitectura de referencia de modelo para el espacio de datos industriales. Fraunhofer en cooperación con la Asociación Industrial de datos espacial. 2017
- [22] NIST Special Publication 800-162. Guía para el atributo de control de acceso basado en Definición y consideraciones (ABAC). Vicente Hu, David Ferraiolo, Rick Kuhn, Adam Schnitzer, Kenneth Sandlin, Robert Miller, Karen Scarfone. De enero de 2014. <http://dx.doi.org/10.6028>
- [23] IEC PAS 63088: Fabricación inteligente - Arquitectura de referencia del modelo de sector 4.0 (RAMI4.0), especificación disposición del público (PAS), la Comisión Electrotécnica Internacional (IEC) de 2017
- [24] Sostenibilidad de formatos digitales: planificación de la Biblioteca del Congreso de Colecciones. Convenciones de empaquetado abierto (Open Office XML), ISO 29500-2: 2008-2012. <https://www.loc.gov/preservation/digital/formats/fdd/fdd000363.shtml>

- [25] La estandarización de Open Office XML. Artículo de Wikipedia.  
[https://en.wikipedia.org/wiki/Standardization\\_of\\_Office\\_Open\\_XML](https://en.wikipedia.org/wiki/Standardization_of_Office_Open_XML)
  
- [26] la estandarización de OpenDocument. Artículo de Wikipedia.  
[https://en.wikipedia.org/wiki/OpenDocument\\_standardization](https://en.wikipedia.org/wiki/OpenDocument_standardization)
  
- [27] El Marco de firma digital de las Convenciones de empaquetado abierto.  
<https://msdn.microsoft.com/en-us/library/aa905326.aspx>
  
- [28] Fundamentos abiertas Convenciones de empaquetado. [https://msdn.microsoft.com/enus/library/windows/desktop/dd742818\(v=vs.85\).aspx](https://msdn.microsoft.com/enus/library/windows/desktop/dd742818(v=vs.85).aspx)
  
- [29] ¿Que es una asignatura digital? Principios fundamentales.  
<http://securityaffairs.co/wordpress/5223/digital-id/what-is-a-digital-signature-fundamentalprinciples.html>
  
- [30] Sostenibilidad de formatos digitales: planificación de la Biblioteca del Congreso de Colecciones. Contenedor de documento del archivo: Core (basado en 6.3.3 postal).  
<https://www.loc.gov/preservation/digital/formats/fdd/fdd000361.shtml>
  
- [31] System.IO.Packaging espacio de nombres. artículo de MSDN. [https://msdn.microsoft.com/enus/library/system.io.packaging\(v=vs.110\).aspx](https://msdn.microsoft.com/enus/library/system.io.packaging(v=vs.110).aspx)
  
- [32] Modelo de Referencia de la norma DIN 16593-1 SPEC para Industrie Arquitecturas 4.0 Servicio - Parte 1: Conceptos básicos de una arquitectura basada en la interacción; Beuth-Verlag: Berlín, Alemania, 2018.  
<https://www.beuth.de/en/technical-rule/din-spec-16593-1/287632675>

## AUTORES

Erich Barnstedt, Microsoft Deutschland GmbH  
 El Dr. Heinz Bedenbender, VDI / VDE-Gesellschaft für Mess- und Automatisierungstechnik (GMA) Meik Billmann, ZVEI -  
 Zentralverband Elektrotechnik- und Elektronikindustrie Dr. Birgit Jefe, Robert Bosch GmbH Erich Clauer, SAP SE Michael Fritsche,  
 IABG

Kai Garrels, ABB STOTZ-KONTAKT GmbH Martin Hankel,  
 Bosch Rexroth AG Oliver Hillermeier, SAP SE

El Dr. Michael Hoffmeister, Festo AG & Co. KG Michael  
 Jochem, Robert Bosch GmbH Dr. Heiko Koziol, ABB AG

El Dr. Christoph Legat, Assystem Germany GmbH Dr. Marco Mendes, Schneider  
 Electric Automation GmbH Dr. Jörg Neidig, Siemens AG Manuel Sauer, SAP SE

Marc Schier, Microsoft Deutschland GmbH Michael Schmitt,  
 SAP SE  
 Tizian Schröder, Otto-von-Guericke-Universität Magdeburg André Uhl, Schneider  
 Electric Automation GmbH Thomas Usländer, Fraunhofer IOSB, Fraunhofer  
 Gesellschaft Thomas Walloschke, Fujitsu Technology Solutions GmbH Bernd Waser,  
 Murrelektronik GmbH Jörg Wende, IBM Deutschland GmbH Constantin Ziesche,  
 Robert Bosch GmbH

Este documento de trabajo ha sido elaborado en el grupo de trabajo “Modelos y Normas” de la ZVEI en colaboración con los Grupos de Trabajo “Arquitecturas de Referencia, reglas y normas” (Plataforma Industrie 4.0), “Seguridad de Sistemas en red”(Plattform Industrie 4,0) y ‘seguridad’(ZVEI).

