

Modelo para la digitalización de activos de la industria batch basado en el estándar ISA 88

Rachel Pairol Fuentes¹[0000-0003-0586-6295], Johnny Alvarado Domínguez¹, Marcela Vegetti¹[0000-0003-4016-1717], Luciana Roldán¹ y Silvio Gonnet¹[0000-0003-3024-4754]

¹ INGAR (UTN - CONICET), 3000 Santa Fe, Argentina
{jaalvarado, rpairol, mvegetti, lroldan, sgonnet}@santafe-conicet.gov.ar

Resumen. Un Asset Administration Shell (AAS) representa digitalmente un asset y facilita la interoperabilidad entre los miembros de la cadena de valor de la Industria 4.0. El AAS organiza la representación de los datos de un asset mediante submodelos compuestos por elementos (datos, propiedades, archivos, colecciones, entre otros) que permiten describir las principales características del asset en diferentes dominios. Diferentes submodelos de componentes electrónicos han sido desarrollados a la fecha, sin embargo, el desarrollo de éstos en el dominio de las industrias de procesos está aún pendiente. Esta contribución propone un modelo para representar digitalmente assets del dominio de la industria de procesos batch. El modelo propuesto parte de la extensión del metamodelo del Asset Administration Shell, e incorpora la definición de submodelos apropiados para este tipo de industria. Los conceptos incluidos están basados en el estándar ISA 88, específicos de la industria de procesos batch. Se presenta también la instanciación de dicho modelo en un caso de uso en la producción de detergente.

Palabras claves: Industria 4.0, RAMI 4.0, ISA 88, Asset Administration Shell.

1 Introducción

La llegada del Internet Industrial de las Cosas (IIoT) ha impulsado la llamada cuarta revolución industrial, Industria 4.0 (I4.0), internet industrial o Smart Manufacturing, según el país donde se origine la iniciativa. Con la ayuda de IIoT, las máquinas, instalaciones, productos y activos en general, son capaces de intercambiar información y controlarse entre sí de forma independiente. Un grupo de instituciones alemanas que integran la denominada Plataforma I4.0 han propuesto un modelo de referencia llamado RAMI 4.0 (Reference Architecture Model for Industrie 4.0) [1].

RAMI 4.0 se basa en un conjunto de estándares organizados en un modelo de 3 dimensiones que describe todos los aspectos críticos de la I4.0. De esta manera, permite descomponer interrelaciones complejas en grupos más pequeños y simples. La integración horizontal y vertical dentro de la I4.0 es esencial. En la RAMI 4.0 la integración horizontal o integración de la cadena de valor, se basa en el estándar IEC 62890 [2]. La integración horizontal permite el intercambio de información acerca del

flujo de material y transacciones entre proveedores, distribuidores y consumidores, brindando transparencia dentro de la cadena de abastecimiento [3]. La integración vertical se logra a través de los niveles de jerarquía (basados en la pirámide de automatización). Estos niveles de jerarquía no necesitan conectarse con sus niveles más cercanos, sino poder conectarse con quien lo requieran, creando así una red de trabajo [4]. Para poder implementar los conceptos de I4.0, es necesario poder compartir información digital dentro de la cadena de valor [5], y para ello, se debe partir de la digitalización de los procesos de producción y los activos que en él intervienen.

Un concepto importante para la RAMI 4.0 es el de Asset Administration Shell (AAS), el cual se define como la representación virtual y la funcionalidad técnica de un objeto que tiene valor para la empresa (un activo o *asset*) [6]. Para estructurar dicha representación digital, se emplean submodelos. Un submodelo define un aspecto específico del asset representado por el AAS. En otras palabras, un AAS está compuesto por distintos submodelos que proveen información de cada dominio técnico de un asset, los cuales pueden ser especificados por medio de diversos estándares. Pueden existir submodelos para identificación, comunicación, y seguridad, entre otros [7]. En particular, la plataforma I4.0 ha publicado submodelos de AAS para: identificación, información de contacto, y placa de identificación del producto. Otros submodelos como: capacidades del asset, mantenimiento, y simulación, se encuentran aún en desarrollo. Dado que el objetivo de la I4.0 está enfocado en lograr la personalización de la producción, y, para ello es indispensable la comunicación entre los distintos dispositivos de la pirámide de automatización, la mayoría de los modelos propuestos a la fecha están enfocados en productos electrónicos como sensores, actuadores, y máquinas CNC (control numérico computarizado), sin embargo, también es posible desarrollar modelos que capturen la información de procesos batch. Las normas (internacionales), especificaciones de consorcios y especificaciones de fabricantes pueden contribuir a la descripción de diferentes submodelos para un AAS. Desde la perspectiva de la interoperabilidad, se busca definir y estandarizar un único submodelo para cada aspecto o dominio técnico.

Existen diferentes tipos de sistemas de producción industrial. En particular, la industria de manufactura por lotes o batch posee características propias, y, por lo tanto, la información a digitalizar también será específica de esta industria. Se ha detectado que actualmente no existen submodelos de AAS que hayan sido propuestos y estén disponibles para ser empleados por este tipo de industria. Sin embargo, se cuenta con la norma ISA 88 [8], que define el modelo de procesos batch y permite especificar, a través de una receta, las actividades que se llevan a cabo en una planta y el orden en que se realizarán, asociando a cada una de estas, los equipos necesarios del modelo físico de la empresa. El objetivo del presente trabajo es proponer un modelo para la definición de AAS para activos propios de la industria de productos fabricados por batch basado en el estándar ISA 88, el cual posibilite representar digitalmente información relativa a la receta de producción, como un paso hacia la digitalización del ciclo de vida del producto.

Este artículo está organizado de la siguiente manera: La sección 2 provee información relevante para entender los conceptos de Asset y Asset Administration Shell, el estándar ISA 88 y trabajos relacionados. La sección 3 introduce el modelo propuesto. La sección 4 presenta la aplicación del modelo propuesto en un caso de estudio. Finalmente, en la sección 5 se muestran las conclusiones y trabajos futuros.

2 Antecedentes y trabajos relacionados

En esta sección se detallan los conceptos más importantes desarrollados por la Plataforma I4.0. La sección 2.1 aborda los conceptos de Asset y Asset Administration Shell. La sección 2.2 describe brevemente el estándar ISA 88, mientras la sección 2.3 detalla algunos de los trabajos relacionados que se han realizado a la fecha.

2.1 Asset y Asset Administration Shell

Un *asset* es todo aquello que tiene valor para la empresa (producto final, inventarios, máquinas), pero también productos intangibles como un software o una idea.

Un *Asset Administration Shell* es la entidad de software que se encarga de la representación digital de un *asset* y permite la interoperabilidad con otros *assets* y entidades localizadas en las capas superiores del modelo RAMI 4 [9]. El AAS almacena la información de un *asset* durante su ciclo de vida: diseño, fabricación, uso, y mantenimiento. También se puede almacenar funciones o complejos algoritmos que permitan la interacción con otros componentes I4.0 [10]. Cada grupo de características del AAS es representado por medio de submodelos, los cuales pueden ser estandarizados, por ejemplo, el submodelo de identificación mediante la ISO 29005, o el de comunicación mediante IEC 61784, entre otros. Si bien, es posible definir submodelos que contengan elementos definidos para un AAS particular, la I4.0 promueve el uso de estándares para soportar la interoperabilidad.

En la Parte 1 de la especificación “Details of the Asset Administration Shell” [7] publicada por Platform I4.0 se presenta un metamodelo para especificar la información de un AAS, el cual se muestra en la Fig. 1.

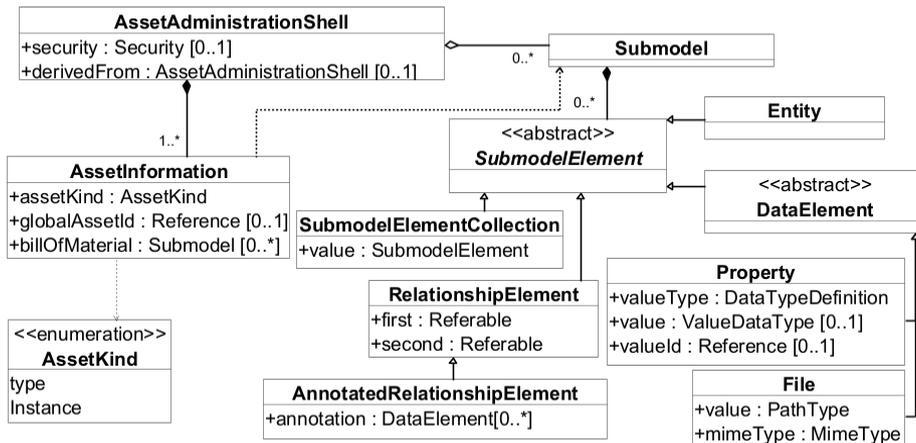


Fig. 1. Metamodelo del Asset Administration Shell. Tomado de [5]

El meta-modelo cuenta con la clase *AssetAdministrationShell* que administra y representa digitalmente la información de los assets de valor para una organización (representado por la clase *AssetInformation*). Un *AssetAdministrationShell* puede tener cero o varios submodelos que describen al *Asset* referenciado según diferentes aspectos dentro de la industria 4.0. Cada submodelo está compuesto por elementos (*SubmodelElement*). Un elemento de submodelo puede representar una colección de elementos (*SubmodelElementCollection* o *SEC*), datos (*DataElement*), propiedades

(*Property*) o archivos (*File*). Un *DataElement* puede estar relacionado con cualquier concepto que sea referenciable (*Referable*), como por ejemplo entidades (*Entity*). Por otro lado, una relación se representa por medio de un *RelationshipElement*. Esta clase se especializa en *AnnotationRelationshipElement*, una clase que permite agregar información sobre la relación entre dos elementos.

2.2 Estándar ISA 88

Según la ISA 88, un proceso de manufactura por batch es aquel que conduce a la producción finita de material (lotes) sometiendo cantidades de materiales de entrada a un orden definido de acciones utilizando uno o más equipos. La programación de un batch requiere de información sobre qué va a ser producido, cuánto va a ser producido, cuándo y en qué orden va a ser producido, así como cuáles equipos van a ser usados. Esta información es provista por medio de recetas.

Dependiendo de la información que requieran los diferentes niveles de la organización, existirán diferentes tipos de recetas. La ISA 88 las clasifica en 4 tipos: una *receta general* especifica materiales, cantidades y los procesos requeridos para la producción de un batch. *La receta de sitio* acomoda la información de la receta general a las políticas, idioma o materia prima propia de un sitio de producción específico. *La receta master* especifica el equipamiento, materia prima y capacidades del proceso o equipo. *La receta de control* es una copia de la receta master que se modifica y (re) programa tanto como sea necesario de acuerdo a información operacional de un batch específico. La receta general y de sitio describen la técnica del proceso, mientras que la receta máster y de control definen el procedimiento para implementar el proceso de acuerdo a los equipos disponibles.

Existen propuestas en la literatura que han abordado la tarea de formalizar los conceptos definidos por la ISA 88. A continuación, se describe parte del modelo conceptual de la ISA 88, que ha sido formalizado e incluido en una de las ontologías propuestas en [11]. En el alcance del presente trabajo, son de interés los conceptos relativos a la representación de la receta y, en particular, la representación de la fórmula incluida en la misma, los cuales se introducen a continuación.

Como se puede ver en la Fig. 2, una receta (*Recipe*) está compuesta por cinco partes representadas por los conceptos cabecera (*Header*), fórmula (*Formula*), requerimientos de equipos (*EquipmentRequirement*), estrategia del procedimiento (*ProcedureSTR*), e información adicional (*OtherInformation*). El *header* de una receta contiene la información administrativa de la receta, como identificación del producto, número de versión, información de aprobación, estado de la receta, y otra información administrativa.

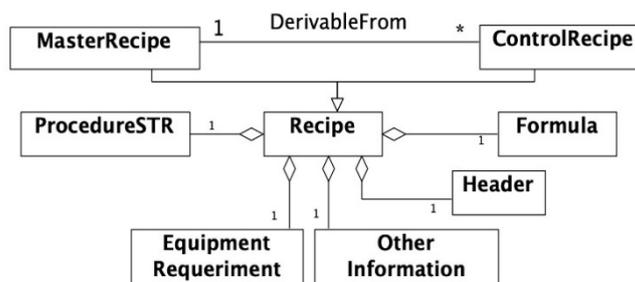


Fig. 2. Principales conceptos del módulo Receta de la ISA 88. Adaptada de [9].

En la Fig. 3 se describe que una fórmula (*Formula*) se relaciona con un conjunto de parámetros (*Parameter*) de un tipo dado (*ValueType*), que pueden ser de entrada, salida o proceso. La información de las entradas del proceso (*InputParameter*) puede ser: la cantidad de materia prima, energía y mano de obra requerida para fabricar el producto. Los parámetros del proceso (*ProcessParameter*) se refieren a información relativa al proceso productivo, tales como temperatura, tiempo, presión, en las distintas etapas de manufactura. Las salidas del proceso (*OutputParameter*) se refieren a los recursos que se obtiene del proceso productivo. Cada parámetro representa la identificación y la cantidad (*Quantity*) de un recurso dado (*Resource*) especificado en la definición de la receta. Estos recursos pueden ser materiales (*Material*), energía (*Energy*), mano de obra (*Manpower*), servicios públicos (*Utility*) y de limpieza (*CIP*), así como entidades de equipo (*EquipmentEntity*) para llevar a cabo actividades de procesamiento, almacenamiento, transferencia y limpieza. La clase *Resource* puede especializarse para considerar otros recursos pertenecientes a dominios industriales específicos.

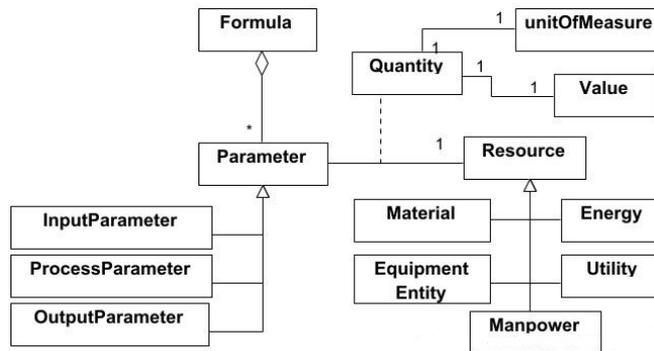


Fig. 3. Modelado de la fórmula de la receta. Adaptado de [9]

Los requerimientos de equipos restringen los equipos que pueden ser usados al implementar una parte específica del procedimiento. El procedimiento define la estrategia para llevar a cabo el proceso de manufactura. La información adicional de la receta puede contener datos referentes a seguridad, empaque, etiquetado o de otros tipos como diagramas de flujo o actividades.

2.3 Trabajos relacionados

A la fecha, se han desarrollado algunos trabajos concernientes a posibles usos del AAS como fuente de información [12], para lograr interoperabilidad entre sistemas actuales [13] [14], para intercambio de datos entre aplicaciones de software [15] y entre aplicaciones de software y dispositivos de control, y para evaluación de riesgos ante ataques cibernéticos [16].

El uso del Asset Administration Shell ha sido enfocado a dispositivos electrónicos y software empresariales; sin embargo, a la fecha no muchos trabajos proponen el uso de AAS para productos de producción masiva. Solo en [17] se propone un AAS de un producto de producción individualizada masiva, como lo es el jugo de fruta. Sin embargo, este trabajo no desarrolla un submodelo para la información de la producción del jugo de fruta, sino que propone un AAS para que un proveedor pueda actualizar la información de la orden de compra en el sistema ERP de su cliente.

Al momento de redactar el presente artículo, los autores no encontraron estudios que propongan un submodelo que permita digitalizar la información de la producción de un batch basado en el estándar ISA 88.

3 Modelo de AAS para la industria de procesos batch

Basado en el estándar ISA 88, se propone un modelo con el objetivo de proveer una estructura de información para la representación digital de Asset Administration Shell relativos al dominio de la industria de producción por batch. Para ello, se extiende el metamodelo de AAS propuesto por la plataforma I4.0 [7], con un modelo genérico para la industria batch que permitirá representar la información de recetas maestras y de control, y de la fórmula de la receta según la especificación ISA 88, que fue introducida en las Fig.2 y Fig.3

Dado que existen diferentes dominios dentro de la industria de procesos batch, se extiende el meta-modelo del AAS de tal manera de mantenerlo flexible, para que pueda ser adaptado a cualquier tipo de industria de procesos batch. En la Fig.4 se presenta el enfoque empleado para la extensión propuesta. Se toma como nivel 1 al Metamodelo del Asset Administration Shell, el cual se extiende con el modelo de procesos de manufactura en lotes, en el nivel 2. Luego, el modelo propuesto en el nivel 2 es instanciado para el dominio particular de la producción de detergente en polvo.

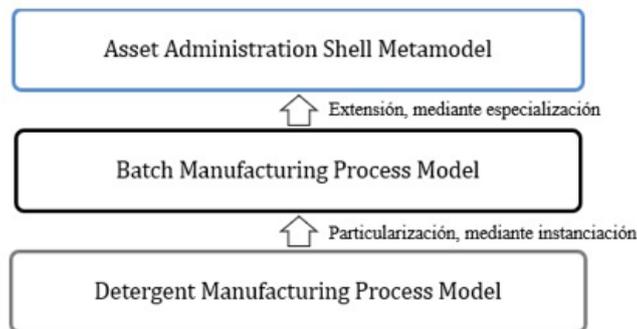


Fig. 4. Propuesta de extensión del metamodelo de AAS

3.1 Propuesta de extensión del metamodelo de AAS mediante especialización

En esta sección se detallan las especializaciones de los principales conceptos del AAS que se muestran en la Fig. 1. Se utiliza notación de estereotipos para indicar los conceptos que se especializan de las clases presentadas en el metamodelo.

En la Fig. 5, como primer paso, se especializa el concepto de *AssetInformation* para representar la información de dos assets genéricos de cualquier industria de producción por lotes: *Producto* (*ProductAssetInformation*) y *Batch* (*BatchAssetInformation*). Para representar la información de Producto, se definen los siguientes submodelos:

1. El submodelo de Datos técnicos (*TechnicalDataSubmodel*), que incluye características técnicas específicas de un producto como olor, color, tamaño, peso, densidad, humedad, sabor.
2. El submodelo de Documentación (*DocumentationSubmodel*), que considera a los elementos que son archivos en pdf, como manuales o especificaciones del producto

Para representar la información del Batch, se definen los siguientes submodelos:

1. El submodelo de proceso de manufactura en lote (*BatchManufacturingProcessSubmodel*).
 2. El submodelo de Datos operacionales (*OperationalDataSubmodel*), que incluye los datos que se registran al momento de la manufactura del producto.
- Estos submodelos están compuestos por colecciones de elementos, que especializan al concepto *SubmodelElementCollection* (se empleará el acrónimo *SEC* cuando se definan nuevos conceptos como especialización de Submodel Element Collection).

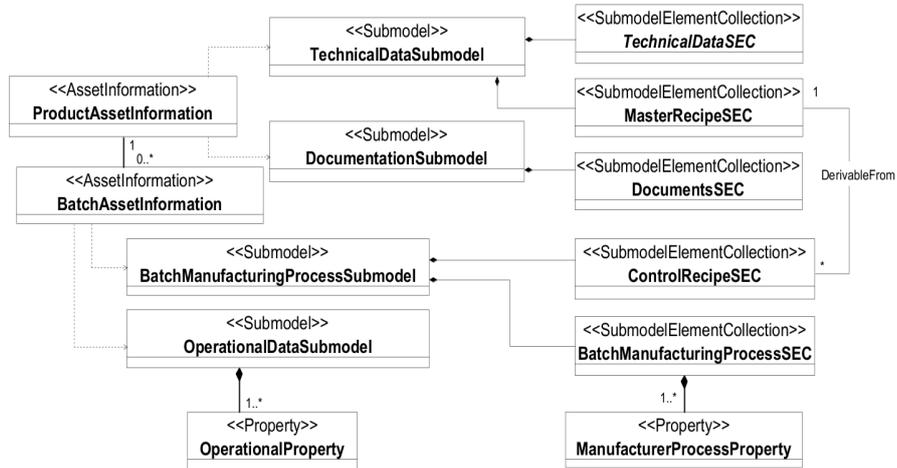


Fig. 5. Especialización del AssetInformation

El *TechnicalDataSubmodel* está conformado por las colecciones de elementos de datos técnicos (*TechnicalDataSEC*) del producto y las colecciones de elementos relativos a la receta máster (*MasterRecipeSEC*) basada en la ISA 88. El *DocumentationSubmodel* está compuesto por una colección de documentos (*DocumentsSEC*) que pueden estar relacionados al diseño del producto.

El *BatchManufacturingProcessSubmodel* está compuesto por las colecciones de elementos *BatchManufacturingProcessSEC*, que contienen propiedades con la información general del proceso de manufactura *ManufacturingProcessProperty* y por *ControlRecipeSEC*, que se deriva de la *MasterRecipeSEC*.

El *OperationalDataSubmodel* contiene un conjunto de propiedades típicas del proceso *OperationalProperty*.

En la Fig. 7 se muestra el *TechnicalDataSEC*, el cual es especializado en: *TechnicalPropertySEC*, en las cuales se relacionan las principales características que determinan un producto, *GeneralInformationSEC*, en las cuales se identifica tanto al productor como al producto; *ProductClassificationSEC*, en la cual se clasifica un producto según determinado estándar (por ejemplo, eclass) y *FutherInformationSEC*, en la cual se puede registrar información adicional que se considere necesaria según cada dominio.

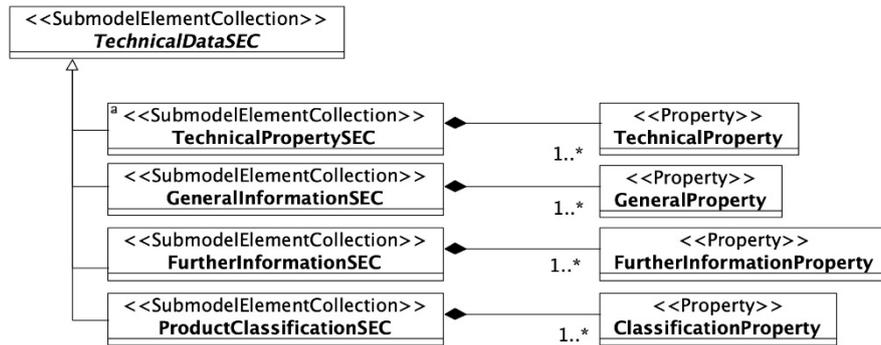


Fig. 6. Especificación para los datos técnicos del producto

Una receta se organiza en componentes (ver Fig. 7), los cuales están representados por *ComponentRecipeSEC*. Esta clase se especializa para representar los distintos componentes que tiene una receta (*FormulaRecipeSEC*, *EquipmentRequirementRecipeSEC*, *ProcedureSTRRecipeSEC*).

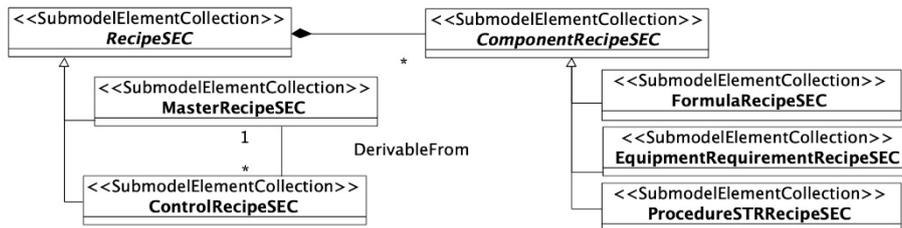


Fig. 7. Especialización del Submodel Element Collection de la receta y sus componentes

Para definir la información que contendrá el AAS de la *FormulaRecipeSEC* (ver Modelado de la fórmula de la receta en Fig. 3), se definen una serie de entidades (*Entity*) para representar los parámetros que la componen (ver Fig. 8).

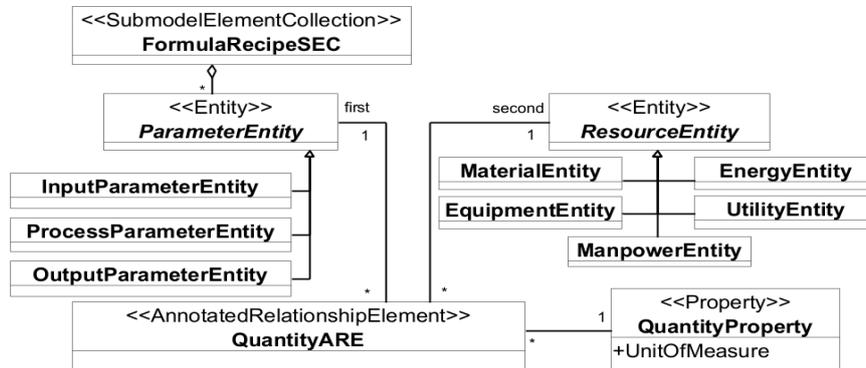


Fig. 8. Especialización del concepto Entity

La entidad *ParameterEntity* abstrae los distintos tipos de parámetros que pueden definirse en una receta: de entrada (*InputParameterEntity*), del proceso

(*ProcessParameterEntity*) y de salida (*OutputParameterEntity*), que están relacionados con las entidades especialmente definidas para los recursos (*ResourceEntity*). Se propone especializar el elemento de submodelo denominado *AnnotatedRelationshipElement* para representar esta relación entre un parámetro y un recurso (*QuantityARE* en la Fig. 9). Por cuestiones de espacio, se empleará el acrónimo *ARE* cuando se definan nuevos conceptos como especialización de *AnnotatedRelationshipElement*. La entidad Recurso (*ResourceEntity*) se especializa para representar los distintos tipos recursos: materiales (*MaterialEntity*), energía (*EnergyEntity*), servicios públicos (*UtilityEntity*), mano de obra (*ManpowerEntity*) y equipamiento (*EquipmentEntity*). La relación *QuantityARE* se vincula con la propiedad que indica la cantidad de un recurso para un parámetro dado (*QuantityProperty*).

3.2 Referenciación de propiedades

Los submodelos y propiedades deben tener un identificador que permita distinguirlos únicamente de los demás elementos del AAS y, permita relacionarlos a definiciones externas como diccionarios o plantillas de submodelos para vincularlos semánticamente. Se recomienda identificar los submodelos con IRI (URI). Las propiedades pueden ser identificadas por medio de (i) IRDI usados en repositorios como ECLASS o en consorcios como IEC o ISO, (ii) IRI, URL o URI (para Ontologías) para propiedades no estandarizadas, pero globalmente únicas [7]. La ontología propuesta en [11], publicada en <https://ontohub.org/scheduling-ontology-network/>, describe los conceptos y relaciones del estándar ISA 88, incluidos en las Fig. 2 y 3. Las URL de dichos conceptos y relaciones son usadas en el presente trabajo como identificadores de los submodelos y propiedades en el caso estudio.

La colección de elementos de submodelo *FormulaRecipeSEC* contiene los parámetros de entrada, salida y de procesos del componente fórmula de la receta. Los parámetros de entrada (*InputParameterEntity*) especificados en la sección 3, tienen como principal propiedad *InputQuantityProperty* (se empleará el acrónimo *IQP* cuando se definan nuevos conceptos como especialización de Input Quantity Property), identificada globalmente como <https://ontohub.org/scheduling-ontology-network/ISA880#InputQuantity>. Esta propiedad indica la cantidad de materia prima o recurso requerido para la fabricación de un batch (*RequiredMaterialIQP*, *RequiredEquipmentIQP*, *RequiredEnergyIQP*, *RequiredUtilityIQP*, *RequiredManpowerIQP*).

Por una cuestión de espacio no se explicitan otras propiedades necesarias para los parámetros. Por ejemplo, con relación a los parámetros de salida, las propiedades a referenciar deberían ser: cantidad de producto por batch (*ProductPerBatchOQP*) e impacto ambiental (*CarbonFootprintGeneratedOQP*) generado en su fabricación. En el caso de los parámetros del proceso se debería referenciar entre otros, el tiempo de procesamiento en un equipo (*ProcessingTimePQP*), presión (*PressurePQP*) y temperatura (*TemperaturePQP*). En la Fig. 9 se muestra una especialización de la *QuantityProperty* para los parámetros de entrada, proceso y salida, teniendo en cuenta los tipos de recursos con los que se vincula. Esta especialización permitiría definir restricciones sobre las unidades de medidas válidas según el tipo de recurso.

Los procesos de fabricación difieren según el tipo de producto que se esté elaborando, por tal razón, las propiedades de las colecciones de elementos de submodelos (*SEC*) *EquipmentRequirementRecipeSEC*, *ProcedureSTRRecipeSEC* y los

parámetros del proceso deben ser referenciadas según las características propias del proceso, por ejemplo, se podría emplear ECLASS.

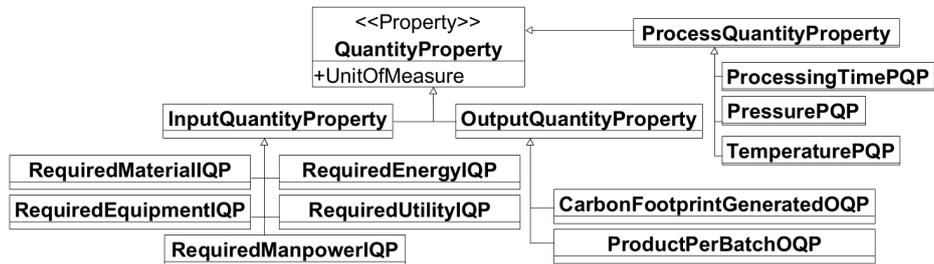


Fig. 9. Especialización para las cantidades de entrada y de salida

4 Caso de Estudio

A continuación, se trabaja con un caso de estudio real que permite validar la factibilidad de instanciar el modelo especializado para generar *AAS* para activos específicos de una industria de procesos batch. Se trata de una empresa que produce detergente en polvo, cuyos clientes son empresas de retailing de grandes superficies. Existe la necesidad de automatizar la recolección de datos del proceso de producción para permitir su trazabilidad y garantizar a sus clientes (empresas de retailing) la calidad del producto adquirido. La organización tiene como objetivo garantizar a los clientes la calidad de su producto mediante la definición de contratos inteligentes basados en Blockchain, que permitan certificar los valores de las características de calidad que son relevantes al cliente, como: cantidad de materia prima, porcentaje de humedad y densidad del producto, certificados de calidad de materias primas, entre otras. Para llegar a este objetivo, el primer paso es la digitalización de sus activos. La producción de esta empresa se realiza en campañas (múltiples batch). En este caso, los activos a digitalizar son el producto “*PretenDetergent*” y la especificación de cada lote de producción “*PretenDetergentBatch*”, los cuales serán instancias de *ProductAssetInformation* y *BatchAssetInformation*, respectivamente.

Para poder validar la aplicación del modelo propuesto al ejemplo presentado se utilizó la herramienta USE (UML Specification Environment), la cual permite validar y verificar especificaciones consistentes de diagramas de clases UML [18]. En la Fig. 10 se muestra parcialmente la instanciación del Asset Administration Shell del detergente *Preten* (*ProductAssetInformation*) y el batch (*BatchAssetInformation*), identificados global y únicamente con el *globalAssetId*.

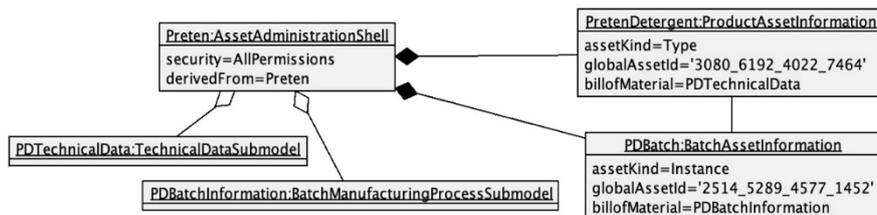


Fig. 10. Instanciación del AAS

El Asset Administration Shell, se compone por los submodelos *PDTechnicalData* y *PDBatchInformation* que contienen la información del detergente y el batch respectivamente. En la Fig. 11 se muestra la instanciación en USE del *TechnicalDataSubmodel*.

El *PDTechnicalData* se compone de *PDMasterRecipe* y *PDTechnicalProperty*. En la *PDMasterRecipe*, se especifica la información de la fórmula. Para la producción del detergente *Preten* se requiere un conjunto de equipos para realizar el proceso. En el caso de estudio presentado estos equipos son: un preparador para mezclar las materias primas, un madurador, luego con la ayuda de una bomba de alta se envía la mezcla a la torre de secado para generar el polvo del detergente que pasa a ser empacado. En la Fig. 11 sólo se incluyen las instancias para los equipos bomba de alta (*PDPumpHigh*) y la torre de secado (*PDDryingTower*) como instancias del *EquipmentRequirementRecipeSEC*.

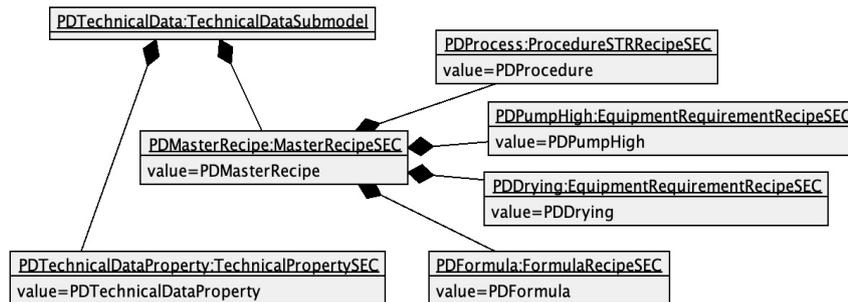


Fig. 11. Instanciación del TechnicalDataSubmodel

La fórmula (*PDFormula*) contiene los parámetros de entrada (*PDInputParameter*). En particular se muestran 3 instancias de *PDInputParameter* que se vinculan con las correspondientes instancias de *QuantityARE*. Por cuestiones de espacio, en la figura se muestran sólo 3 de los materiales necesarios para la producción del detergente. En particular, se muestran las relaciones que indica que se requieren 136kg de ácido sulfúrico (*SulfuricAcid*), 277 Kg de silicato de sodio (*SilicatoSodio*) y 34,7 kg de soda caustica (*CausticSoda*). (ver Fig. 12).

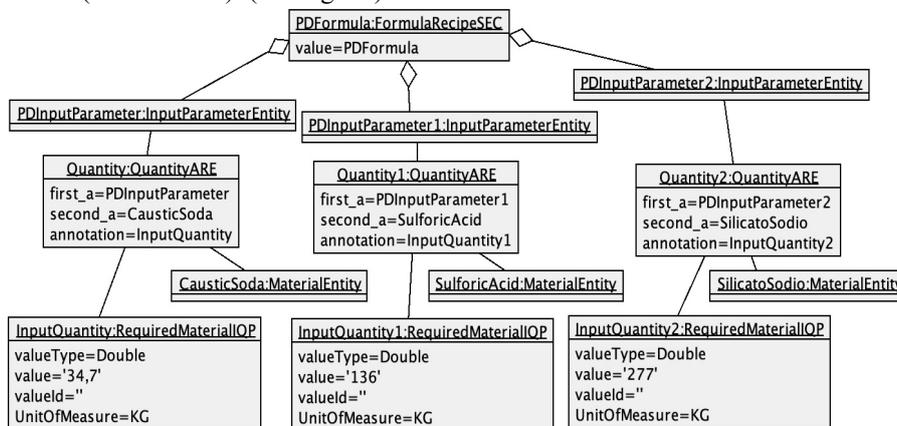


Fig. 12. Parámetros de entrada, materiales y cantidades de la fórmula

Teniendo en cuenta el objetivo que tiene la empresa de la definición de contratos inteligentes que garanticen a los clientes los valores de densidad, humedad, peso y olor de los productos fabricados, se agregan al modelo de la Fig. 13, cuatro instancias de *TechnicalDataProperty*, que permiten representar estos valores (*Humidity*, *Density*, *Weight* y *Odor*).

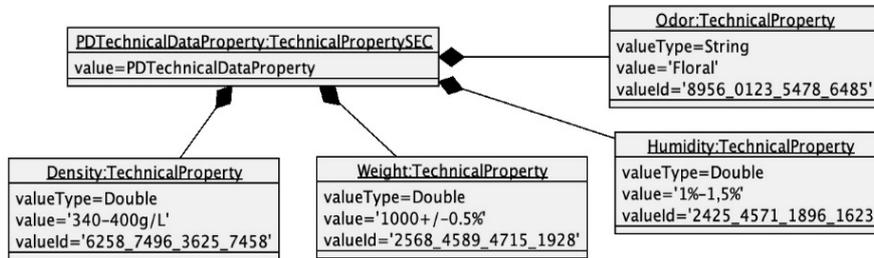


Fig. 13. Datos técnicos

La información relevante del batch (*PDBatchInformation*) se instancia como se muestra en la Fig. 14.

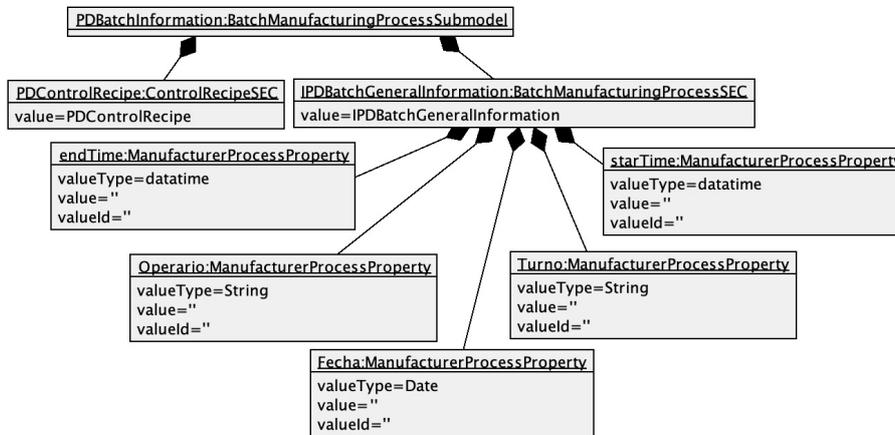


Fig. 14. Instanciación del BatchManufacturingProcessSubmodel

Por medio del empleo de la herramienta USE (UML Specification Environment) se valida y verifica la consistencia de las especificaciones del submodelo propuesto [18].

5 Conclusiones

En el presente trabajo se presentó un modelo para representar digitalmente assets del dominio de la industria de procesos batch. El modelo propuesto parte de la extensión del metamodelo del Asset Administration Shell propuesto por la Plattform Industrie 4.0 alemana, e incorpora la definición de submodelos apropiados para este tipo de industria. Los conceptos incluidos, los cuales son específicos de la industria de procesos batch, están basados en el estándar ISA 88, lo que garantiza que se emplea el vocabulario normalizado propio de la industria que ha sido acordado a través del esfuerzo de numerosas organizaciones.

El modelo incluye los elementos necesarios para la representación digital de dos assets fundamentales de esta industria: el Producto y el Lote. Para la caracterización de éstos se definieron 4 submodelos para diferentes aspectos de información: los datos técnicos, y de documentación para el producto, y los datos de proceso de manufactura y datos operacionales para el Lote. El submodelo de datos técnicos abarca los elementos que forman parte de la receta maestra, mientras que el submodelo de manufactura en lote abarca los elementos para digitalizar las recetas de control y las características del proceso batch.

El modelo propuesto puede ser instanciado para diferentes tipos de producción batch. En esta propuesta se presentó la instanciación para un proceso batch de producción de detergente en polvo, en el cual se indicaron específicamente las materias primas, equipos e información relacionada al batch.

La plataforma i4.0 cuenta con una herramienta denominada AASX Package Explorer¹, para la creación de assets administration shells. Esta herramienta implementa el metamodelo básico de AAS y por lo tanto cuenta con los elementos mínimos para crear representaciones digitales de los assets. El modelo propuesto sentaría las bases conceptuales para la implementación de una herramienta para la definición de AAS específicos para el dominio de la industria de manufactura en lote. De esta manera, la herramienta ofrecería al usuario los elementos con la semántica apropiada y propiedades específicas para facilitar la creación de los AAS. Se propone como trabajo a futuro trabajar en la extensión de dicha herramienta en base al modelo propuesto.

6 References

- [1] “Reference Architecture Model Industry 4.0 (RAMI 4.0). 91345:2016-04,” 2016, [Online]. Available: <https://www.beuth.de/de/technische-regel/din-spec-91345/250940128>.
- [2] International Electrotechnical Commission (IEC), “Industrial-process measurement, control and automation – Life-cycle management for systems and components (IEC 62890),” 2020.
- [3] M. Asif, C. Searcy, and P. Castka, “Exploring the role of industry 4.0 in enhancing supplier audit authenticity, efficacy, and cost effectiveness,” *J. Clean. Prod.*, vol. 331, no. December 2021, 2022, doi: 10.1016/j.jclepro.2021.129939.
- [4] C. Kaar, J. Frysak, and C. Sary, “Scaffolding RAMI4.0-Exploration as Design Support,” *ACM Int. Conf. Proceeding Ser.*, 2018, doi: 10.1145/3232078.3232098.
- [5] J. M. Müller, J. W. Veile, and K. I. Voigt, “Prerequisites and incentives for digital information sharing in Industry 4.0 – An international comparison across data types,” *Comput. Ind. Eng.*, vol. 148, 2020, doi: 10.1016/j.cie.2020.106733.
- [6] Plattform Industrie 4.0, “Status Report - RAMI4.0,” *ZVEI – Ger. Electr. Electron. Manuf.*, vol. 0, no. July, p. 28, 2015, [Online]. Available: https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikation

¹ <https://github.com/admin-shell-io/aasx-package-explorer>

- en/2016/januar/GMA_Status_Report_Reference_Architecture_Model_Industrie_4.0_RAMI_4.0/GMA-Status-Report-RAMI-40-July-2015.pdf.
- [7] S. Bader, E. Barnstedt, H. Bedenbender, M. Billman, B. Boss, and A. Braunmandl, "Details of the Asset Administration Shell Part 1 - The exchange of information between partners in the value chain of Industrie 4.0," *Plattf. Ind. 4.0*, vol. 0, p. 473, 2020.
- [8] ISA, Batch Control, Part 1: Models and Terminology, ISA-88.01-1995 (R2006). 2006.
- [9] F. Siqueira and J. G. Davis, "Service Computing for Industry 4.0: State of the Art, Challenges, and Research Opportunities," *ACM Comput. Surv.*, vol. 54, no. 9, 2022, doi: 10.1145/3478680.
- [10] I. Grangel-gonzález, L. Halilaj, S. Auer, S. Lohmann, C. Lange, and D. Collarana, "An RDF-based Approach for Implementing Industry 4 . 0 Components with Administration Shells."
- [11] M. Vegetti and G. Henning, "Journal of Industrial Information Integration Ontology network to support the integration of planning and scheduling activities in batch process industries," *J. Ind. Inf. Integr.*, vol. 25, no. January 2020, p. 100254, 2022, doi: 10.1016/j.jii.2021.100254.
- [12] D. Lang, S. Grunau, L. Wisniewski, and J. Jasperneite, "Utilization of the asset administration shell to support humans during the maintenance process," *IEEE Int. Conf. Ind. Informatics*, vol. 2019-July, pp. 768–773, 2019, doi: 10.1109/INDIN41052.2019.8972236.
- [13] M. A. Inigo, A. Porto, B. Kremer, A. Perez, F. Larrinaga, and J. Cuenca, "Towards an Asset Administration Shell scenario: A use case for interoperability and standardization in Industry 4.0," *Proc. IEEE/IFIP Netw. Oper. Manag. Symp. 2020 Manag. Age Softwarization Artif. Intell. NOMS 2020*, 2020, doi: 10.1109/NOMS47738.2020.9110410.
- [14] X. Zhang, "An Industry 4 . 0 Asset Administration Shell-Enabled Digital Solution for Robot-Based Manufacturing Systems," *IEEE Access*, vol. 9, pp. 154448–154459, 2021, doi: 10.1109/ACCESS.2021.3128580.
- [15] X. Ye, M. M. Yu, W. S. Song, and S. H. Hong, "An Asset Administration Shell Method for Data Exchange Between Manufacturing Software Applications," *IEEE Access*, vol. 9, pp. 144171–144178, 2021, doi: 10.1109/ACCESS.2021.3122175.
- [16] W. Kastner, "A Centralised or Distributed Risk Assessment using Asset Administration Shell."
- [17] J. H. Hang, Y. M. Lim, and G. S. Bhd, "Towards Mass Individualized Production: RAMI 4 . 0 Asset Data Channelling for Manufacturing Value Chain Connectivity," vol. 0, pp. 225–231, 2021.
- [18] M. Gogolla, F. Büttner, and M. Richters, "USE: A UML-based specification environment for validating UML and OCL," *Sci. Comput. Program.*, vol. 69, no. 1–3, pp. 27–34, 2007, doi: 10.1016/j.scico.2007.01.013.