

An Asset Administration Shell Submodel for Representing the Procedural Part of ISA-88 Recipes

Johnny Alvarado Domínguez , Marcela Vegetti , and Silvio Gonnet 

Abstract— It is undeniable the benefits that the implementation of digital twins provides to industries. However, the greatest advances in this regard have been made in the definition and implementation of digital twins in discrete manufacturing industries. The development of these twins is still in its early stages in process industries. An important issue in creating digital twins to support decision-making in the process industry is to be able to describe the production procedures. This paper aims to present an *Asset Administration Shell* submodel that allows the representation of procedural recipes in the batch process industry based on the ISA-88 standard. This paper proposes a conceptual model to represent the Sequential Function Chart language, which is one of the languages proposed by the mentioned standard to represent manufacturing procedure. In addition, the proposal includes a set of rules to map *Sequential Function Chart* concepts into concepts belonging to the *Asset Administration Shell* metamodel introduced by Platform Industrie 4.0. These mapping rules would allow the implementation of tools that automatically translate existing *Sequential Function Chart* models into *Asset Administration Shell* submodels to reuse existing knowledge for the implementation of digital twins in batch process industries.

Link to graphical and video abstracts, and to code: <https://latamt.ieer9.org/index.php/transactions/article/view/9224>

Index Terms— Asset Administration Shell, Digital Twin, ISA-88, procedure submodel, Sequential Function Chart.

I. INTRODUCTION

AS empresas se enfrentan a la tendencia de la personalización de la producción, caracterizada por una disminución del tamaño de lote, mayor variedad de productos y procesos de producción más sofisticados, dificultando la planeación y control de la producción (PCP) [1]. El uso de tecnologías como los gemelos digitales podría soportar las tareas de PCP en las industrias [2], [3]. Para ello, es necesario poder integrar la información proveniente de distintas máquinas y procesos que generalmente se encuentran en distintos formatos y lenguajes, ocasionando un problema de interoperabilidad. La plataforma Industria 4.0 ha propuesto el concepto de *Asset Administration Shell* (*AAS*), considerado la

The associate editor coordinating the review of this manuscript and approving it for publication was Guillermo Valencia-Palomo (*Corresponding author: Johnny Alvarado*).

This work was supported by CONICET, Project PIP11220210100305CO, and Universidad Tecnológica Nacional, Project UTI8599TC.

J. Alvarado Domínguez is with Asociación de Investigación Metalúrgica del Noroeste, Porriño, Spain (e-mail: johnny.alvarado@aimen.es).

M. Vegetti, and S. Gonnet are with INGAR, Avellaneda, Santa Fe, Argentina (e-mails: mvegetti@santafe-conicet.gov.ar, and sgonnet@santafe-conicet.gov.ar).

piedra angular de la interoperabilidad, cuya implementación se materializa a través de un gemelo digital [4].

El *AAS* es usado como un modelo común de información en la planta y entre actores de la cadena de valor, describiendo un *asset* por medio de submodelos que hacen referencia a un dominio específico del mismo [4]. Aunque los gemelos digitales han sido propuestos ampliamente en las industrias de manufactura discreta [5], [6], no sucede lo mismo en las industrias de procesos. Además, el uso de gemelos digitales podría soportar las tareas de planeación y programación de la producción [3], [7] y mejorar la eficiencia de los sistemas de producción [8] en la industria de procesos. La industria de procesos puede involucrar manufactura continua, por *batch* o una combinación de ambas [9]. En los procesos *batch*, se producen cantidades finitas de material mediante actividades programadas en un tiempo determinado y utilizando varios equipos [9]. El diseño de un gemelo digital para este tipo de industria requiere la definición de submodelos que capturen la información de los distintos procesos. Los estándares proveen conceptos y relaciones que permiten describir un dominio específico [4]. El estándar ISA-88, ampliamente usado en la industria de procesos, permite especificar los procesos de producción, a través de la definición de recetas donde se determina qué, cómo, cuándo, en qué orden y con qué recursos se llevará a cabo la producción de cada *batch* [9]. Una de las partes de la receta es el procedimiento, el cual permite especificar las acciones del proceso previstas, así como las operaciones y etapas del proceso de manufactura. Uno de los lenguajes sugeridos por el mencionado estándar para la representación de procedimientos es el *Sequential Function Chart* (*SFC*). Este lenguaje tiene las características para representar la lógica presente en los procedimientos llevados a cabo por los equipos de lógica y control, como los PLC [10].

En [11] los autores introducen la especificación de un submodelo de *AAS* basado en el estándar ISA-88. Sin embargo, dicha propuesta no aborda la representación de la estructura procedimental de la receta, y el submodelo queda definido únicamente a nivel conceptual, sin considerar su implementación. El objetivo de este trabajo es presentar la definición de un submodelo de *AAS* que permite la representación digital de procesos productivos en la industria de procesos *batch*, basado en un modelo conceptual del lenguaje *SFC*. Asimismo, se introduce un conjunto de reglas que permiten transformar este modelo conceptual especificado en UML (*Unified Modeling Language*) en una instancia del metamodelo de *AAS*. La definición de estos mapeos sienta las bases para desarrollar herramientas automatizadas que permitan la conversión de modelos *SFC* en modelos compatibles con herramientas de Industria 4.0, como el *AASX*.

Package Explorer. Así, se evita que los expertos en un dominio industrial deban familiarizarse con los conceptos específicos de Industria 4.0 para definir la representación digital de un proceso. Para desarrollar estas herramientas, se propone un enfoque de Ingeniería Dirigida por Modelos (MDE - Model Driven Engineering) [12], en el cual modelos conceptuales de alto nivel son transformados en artefactos de software mediante reglas de mapeo. Así, disponiendo de i) el modelo conceptual de SFC, un lenguaje conocido por los expertos del dominio industrial, y ii) los mapeos entre los conceptos de este modelo y los conceptos del metamodelo de AAS, es posible, siguiendo un enfoque MDE, la construcción automatizada de herramientas para la Industria 4.0

La implementación del submodelo propuesto permite la descripción de un procedimiento de manufactura en un lenguaje legible por máquina, tales como JSON, XML o RDF, lo que permite la creación de gemelos digitales de los procesos *batch*. Al proporcionar una representación virtual precisa de los procesos *batch* en tiempo real, los gemelos digitales permiten soportar la toma de decisiones en las actividades de planeación y control de la producción.

Este artículo se estructura de la siguiente manera: la Sección 2 aborda los conceptos de AAS, el estándar ISA-88, el lenguaje SFC y trabajos previos relacionados; la Sección 3 describe la propuesta, incluyendo el modelo conceptual, su aplicación en la construcción del AAS y una prueba de concepto en la herramienta AASX Package Explorer; la Sección 4 presenta una discusión de la propuesta; y la Sección 5 expone las conclusiones y los desafíos futuros.

II. ANTECEDENTES

Esta sección introduce los principales conceptos utilizados en las siguientes secciones del trabajo. En la sección A se discuten los conceptos relacionados a *AAS*. En la sección B se presentan los conceptos relacionados al estándar ISA-88 y el lenguaje *SFC*, mientras que la sección C detalla trabajos que a la fecha han desarrollado *AAS* y gemelos digitales en la industria de procesos.

A. Asset Administration Shell

El concepto central de la I4.0 es el *Asset*. Un *asset* es cualquier objeto de valor para una organización (máquinas, sensores, software, productos). El propósito de la industria 4.0 es facilitar la cooperación y colaboración entre *assets*, los cuales deben estar conectados y representados virtualmente [13]. La Fig. 1 presenta una versión resumida del metamodelo de *AAS* [4]. El *AAS* permite la descripción de un *asset* mediante submodelos. Cada submodelo (*Submodel*), se compone de elementos de submodelo (*SubmodelElement*), que refieren a un dominio específico del *asset*. Existen distintos tipos de elementos que pueden incluirse en un submodelo, por ejemplo, propiedades (*Property*), relaciones (*RelationshipElement*), listas (*SubmodelElementList*) o colecciones (*SubmodelElement-Collection*) de estos elementos. La definición de submodelos basados en estándares permite la creación de plantillas de submodelos y garantiza la

interoperabilidad [4]. El estándar ISA-88 [9] provee una guía para el diseño y especificaciones de sistemas de control *batch*.

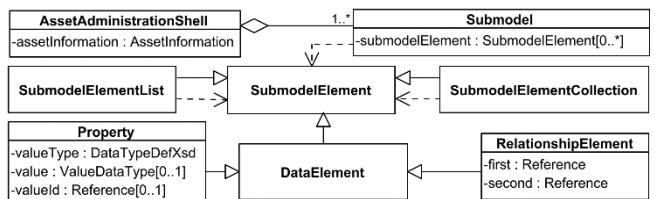


Fig. 1. Resumen del Metamodelo de *AAS* [4].

B. Estándar ISA-88 y Lenguaje Sequential Function Chart

Según el estándar ISA-88, una receta se organiza en las siguientes partes: encabezado, fórmula, requerimiento de equipos, estructura de procedimiento y otra información [9]. La receta se puede definir en distintos niveles: procedimiento, unidad de procedimiento, operación, y fase. La propuesta presentada en este trabajo se enfoca en el nivel de fase. El estándar ISA-88 reconoce que el lenguaje *SFC* puede ser usado para representar la lógica de un procedimiento a nivel de fase. El lenguaje *SFC* es uno de los lenguajes del estándar IEC 61131-3 [14] para la especificación de programas en controladores lógicos. Es un lenguaje gráfico y, en consecuencia, no entendible por las computadoras. Lo que impide la detección automática de especificaciones de procesos no válidos. Los conceptos principales representados mediante el lenguaje *SFC* son explicadas en la parte A de la sección III.

C. Trabajos Relacionados

El uso de gemelos digitales que soporten las tareas de PPC ha ganado interés en el ámbito académico en los últimos años. En [7] se presenta un concepto de gemelos digitales que soporte las actividades de PPC en un caso de uso de moldeado de plástico por inyección. En este trabajo, el gemelo digital le brinda soporte al programador de la producción basado en datos propios del sistema de manufactura. En [15] se propone el uso de *AAS* para la integración horizontal entre miembros de una cadena de valor en un contexto de producción personalizada en masa. Sin embargo, la propuesta fue desarrollada *ad-hoc*, sin seguir algún estándar que permitiera la interoperabilidad con otros actores de la cadena de valor. Los autores en [3] consideran que la implementación de un gemelo digital basado en la receta soportaría las actividades de simulación y programación de la producción. Sin embargo, ese trabajo sólo fue una propuesta que pretendía mostrar los beneficios de un gemelo digital en la industria alimenticia.

Las investigaciones sobre el uso de *AAS* en aplicaciones industriales han ganado interés en los últimos años. Los autores en [16], [17], [18] implementan soluciones usando *AAS* para el monitoreo y predicción del consumo de energía en ambientes de producción. En [19] los autores proponen un mapeo del *AAS* y el modelo de función de bloque del estándar IEC 61499 con el fin de combinar la información estática (provista por el *AAS*) y la información dinámica (provista por el *asset*) y posibilitar la falta de flexibilidad y reconfiguración dinámica en un ambiente de producción distribuida. En [20]

los autores realizan un mapeo de los conceptos de la configuración de programas IEC 61131 (*Resource*, *POUs*, *Task*) y el *AAS* para un PLC que facilite una reconfiguración de los procesos. Estos dos últimos trabajos demuestran que es posible poder especificar la lógica inherente a los procesos de fabricación mediante *AAS* empleando lenguajes especificados en estándares como el IEC 61131 o el IEC 61499. En [21] los autores identifican a los estándares ISA-88 e IEC 61131 como relevantes para la definición de plantillas de submodelos. Sin embargo, no desarrolla algún submodelo basado en ellos. El presente trabajo tiene como objetivo la creación de un submodelo de *AAS* para la representación de procedimientos de fabricación *batch* basado en el lenguaje *SFC*, el cual es uno de los lenguajes descritos en el estándar IEC 61131, lo que permitiría, una vez estandarizado, garantizar la interoperabilidad de los datos generados en planta.

III. PROPUESTA DE SUBMODELO DE *AAS* BASADO EN *SFC*

Esta sección se divide de la siguiente forma: En la subsección A se describe la conceptualización del modelo *SFC*, incluyendo reglas para detectar redes inseguras e inalcanzables. La subsección B, se muestra la especificación del submodelo de *AAS* basado en la formalización propuesta. Por último, la subsección C, se introduce la implementación del submodelo en la herramienta *AASX Package Explorer*. La Fig. 2 muestra el proceso para llevar a cabo la propuesta del submodelo de *AAS* basado en *SFC*.

A. Conceptualización del Lenguaje *SFC* y Procedimiento para la Detección de Redes Inseguras e Inalcanzables

En esta sección se definen los conceptos principales del lenguaje *SFC* y su representación en el modelo conceptual propuesto. El elemento principal del *SFC* es la red, compuesta por pasos (*Step*) y transiciones (*Transition*), generalizados como nodos (*Node*) en la propuesta (Fig. 3). Un *Step* tiene los atributos: nombre (*name*), que lo identifica, un *stepFlag* de tipo booleano que indica si el paso está activo, y un *elapsedTime* que muestra el tiempo de activación. Una *Transition* tiene los atributos: *priority*, que controla la precedencia en la evaluación de transiciones, y *transitionCondition*, una expresión booleana para el cambio de estado de un *Step*. Cada red comienza con un paso inicial (*InitialStep*). Los enlaces (*Links*) en una red *SFC* son asociaciones entre nodos, mostradas en la Fig. 3. El prefijo *orig* denota el nodo predecesor y *end* el sucesor. Por ejemplo, la asociación en la parte inferior de la Fig. 3, con *origTS* y *endTS*, indica que el enlace va de una transición predecesora a un paso sucesor. Una secuencia se compone de pasos y transiciones alternados. *SFC* define: i) secuencias simultáneas, que permiten pasos en paralelo, y ii) secuencias divergentes, que especifican flujos alternativos (solo una secuencia puede ejecutarse). El modelo incluye clases para indicar el inicio (*InitialofSimultaneousSequence* y *InitialofDivergenceSequence*) y fin (*ConvergenceofSimultaneousSequence* y *ConvergenceofSequence*) de estas secuencias. También se definen ciclos (*Loop*) para enlazar una transición a un paso

previo. Las secuencias y *Loop* se generalizan en la clase *Control* (Fig. 4a), que es una especialización de *Node*.

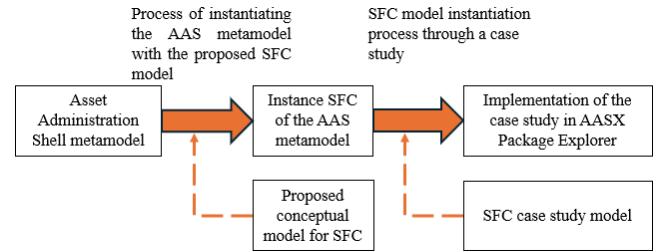


Fig. 2. Proceso para la propuesta de submodelo de *AAS*.

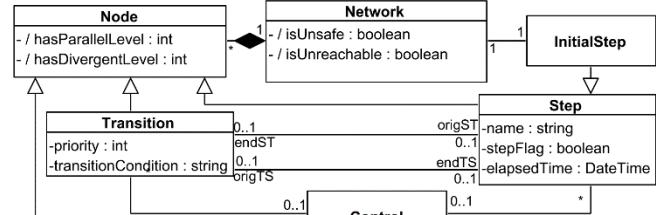


Fig. 3. Modelo conceptual propuesto del lenguaje *SFC*.

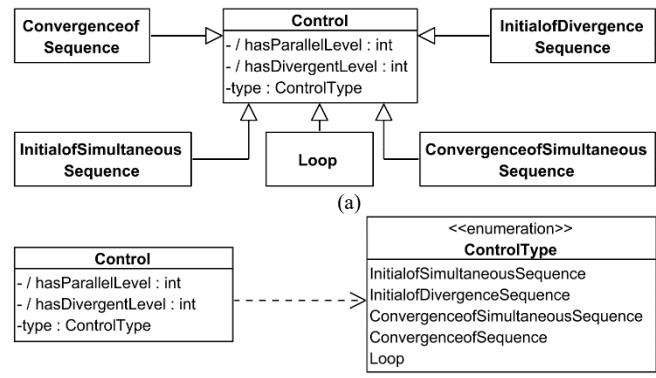


Fig. 4. (a). Especialización de la clase *Control*. (b). Representación alternativa empleando atributo para definir la dimensión *type*.

La Fig. 4b muestra una representación alternativa adoptada en este trabajo, en la que se transforma la especialización en un atributo de dimensión explícita, aplicable cuando las especializaciones no tienen propiedades particulares [22].

Es posible representar redes inseguras y redes inalcanzables con *SFC* [23]. Las redes inseguras son aquellas que permiten que dos o más pasos no simultáneos sean ejecutados al mismo tiempo. Las redes inalcanzables son aquellas que no logran ser ejecutadas en su totalidad. Las redes inseguras e inalcanzables se generan cuando transiciones que están dentro de una secuencia en paralelo o divergente son enlazadas con pasos que están por fuera de dichas secuencias. El uso de un lenguaje gráfico usado por las herramientas para definir modelos *SFC* dificulta la detección de este tipo de redes no válidas. Para resolver esto, se complementó la especificación del modelo conceptual en UML con el empleo del lenguaje de restricciones OCL (*Object Constraint Language*). El modelo conceptual resultante fue especificado en el ambiente USE (*UML-Based Specification Environment*) [24], donde fue posible validar la detección de redes *SFC* inválidas. Las reglas

para la detección de las redes inseguras e inalcanzables fueron creadas con la idea de identificar si un *Node* pertenece a una secuencia paralela o a una secuencia divergente. Para ello, se agregaron a la clase *Node* los atributos derivados *hasParallelLevel* y *hasDivergentLevel* (Fig. 3), los cuales no se encuentran definidos en la especificación del lenguaje *SFC*. Luego, se propusieron reglas de derivación OCL para la especificación de los valores de estos. En la Fig. 5 se muestra la regla de derivación del atributo *hasParallelLevel* para la clase *Step*. De igual forma se muestra la regla para la derivación del atributo *hasParallelLevel* para el nodo inicio de secuencia simultánea.

Para el caso del nodo llamado *Step*, la regla indica que todo paso inicial (*InitialStep*) tiene un valor para el atributo *hasParallelLevel* igual a 0. Cuando un paso (no inicial) es precedido por una transición (*origTS->notEmpty()*), un *loop* (*origLoopS->notEmpty()*), una convergencia de secuencia divergente (*origCOSS->notEmpty()*), o un inicio de secuencia simultánea (*origISSS->notEmpty()*) el valor para el atributo *hasParallelLevel* del paso es igual al del nodo que lo precede. Para el caso del inicio de secuencia simultánea, la regla de derivación del atributo *hasParallelLevel* especifica que el valor de dicho atributo se calcula sumando 1 al valor del atributo *hasParallelLevel* que tiene la transición que le precede. Si la transición que le precede tiene como valor del atributo *hasParallelLevel* igual a 0, quiere decir que no hay una ejecución en paralelo previo y el valor que tomará será igual a 1, indicando que se ha empezado una ejecución en paralelo. Si, por el contrario, la transición que le precede tiene un valor diferente de cero, por ejemplo 1, el valor para el atributo *hasParallelLevel* del inicio de secuencia simultánea será 2, indicando que se está ejecutando una operación en paralelo, dentro de otra operación en paralelo y que, para evitar problemas de redes inseguras o inalcanzables, se deben cerrar dichas ejecuciones en paralelo (mediante fin de secuencias simultáneas). El objetivo es poder identificar que los nodos solo se enlazan con otros nodos que tienen el mismo valor de atributo *hasParallelLevel* y *hasDivergentLevel*. Para los otros tipos de nodos se definieron también las correspondientes especificaciones OCL para inferir los valores de los atributos *hasParallelLevel* y *hasDivergentLevel*.

También, se definieron los atributos *isUnsafe* e *isUnreachable*, en la clase *Network*. Estos atributos son de tipo booleano y de acuerdo con la evaluación de las reglas OCL definidas (Fig. 5) determina si la *Network* es insegura/inalcanzable (*true*) o no (*false*). El atributo *isUnsafe* del concepto *Network*, es *true* si existiendo un *Loop*, las transiciones que participan en el *Loop* tienen distintos valores para el atributo *hasParallelLevel* o *hasDivergentLevel*. Esta diferencia indica que las transiciones que participan en el *Loop* pertenecen a secuencias paralelas o divergentes que no han sido finalizadas. De igual forma, la regla del atributo *isUnreachable* evalúa *true* si existiendo un *Loop* en la *Network*, el valor del atributo *hasParallelLevel* o el valor del atributo *hasDivergentLevel* del *Loop* es un número negativo.

B. Construcción del modelo de AAS

A partir de la conceptualización detallada en la sección previa, se instancia el metamodelo de AAS, obteniendo un submodelo basado en el estándar ISA-88. Como se indicó en la sección III.A, el lenguaje *SFC* permite especificar un procedimiento a nivel de fase mediante redes (*Network* en la Fig. 3), representadas en esta propuesta como instancias de *SubmodelElementCollection*. Además, los pasos, transiciones y controles se representan mediante colecciones denominadas *StepContainer*, *TransitionContainer* y *ControlContainer*, agrupadas en *StepSet*, *TransitionSet*, y *ControlSet*, respectivamente. El metamodelo de *AAS* no permite la

```

class Step < Node
attributes
hasParallelLevel: Integer derive: if self.oclisTypeOf(InitialStep) then 0 else
if self.origLoopS->notEmpty() then self.origLoopS.hasParallelLevel else
if self.origTS->notEmpty() then self.origTS.hasParallelLevel else
if self.origCOSS->notEmpty() then self.origCOSS.hasParallelLevel
else self.origISSS.hasParallelLevel endif endif endif endif
end

class InitialofSimultaneousSequence < Control
attributes
hasParallelLevel: Integer derive: if self.origTISS->notEmpty() then
self.origTISS.hasParallelLevel + 1 else self.origTISS.hasParallelLevel
endif
end

class Network
attributes
isUnsafe: Boolean derive: self.control->select(c| c.oclisTypeOf(Loop))-
->collect(c|c.oclisTypeOf(Loop))->exists(|| l.origTLoop ->
exists(t1, t2| t1.hasParallelLevel <> t2.hasParallelLevel))or self.control-
->select(c| c.oclisTypeOf(Loop)) ->collect(c|c.oclisTypeOf(Loop))->exists(|| l.origTLoop -
->exists(t1, t2| t1.hasDivergentLevel <> t2.hasDivergentLevel))

isUnreachable: Boolean derive: self.control->select(c| c.oclisTypeOf(Loop))-
->collect(c|c.oclisTypeOf(Loop))->exists(|| l.hasParallelLevel <0 or l.hasDivergentLevel<0)
end

```

Fig. 5. Ejemplos de reglas de derivación de atributos en OCL.

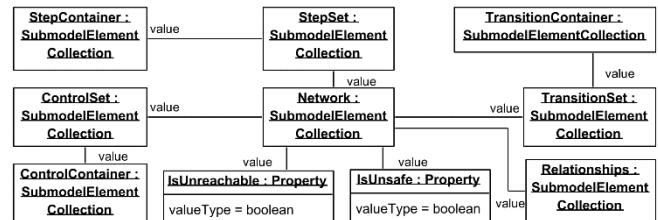


Fig. 6. Representación de un Network SFC.

representación de composición utilizada en el modelo UML, por lo que estas composiciones se instancian como la propiedad *value*, como se observa en la Fig. 6. También se incluyen en la clase *Network* las relaciones entre nodos, representadas mediante la colección *Relationships*.

Los atributos *isUnsafe* e *isUnreachable* de la clase *Network* (Fig. 3) se representan como instancias del concepto *Property* del metamodelo *AAS* (Fig. 1), vinculadas a la *Network* por medio de la asociación *value*. Los atributos de cada nodo se agrupan en *SubmodelElementCollection*. Un *StepContainer* (Fig. 7) incluye los atributos: *elapsedTime*, *stepFlag*, *hasDivergentLevel*, *hasParallelLevel* y *name* (Fig. 3), como *properties*. En el modelo conceptual (Fig. 3), *Step* se especializa en *InitialStep*, pero sin propiedades particulares, por lo que se incluyó como propiedad *InitialStep*, de tipo booleano, en *StepContainer* (Fig. 7).

El metamodelo de *AAS* no soporta la representación de relaciones de generalización/especialización, por lo que no es

possible representar clases abstractas como *Node* (Fig. 3) para luego especializarlas. Así, en el submodelo se debe repetir la definición de propiedades en sus clases concretas. Para los nodos de control, se optó por un enfoque diferente. Dado que las especializaciones de *Control* (Fig. 4a) no añaden nuevas propiedades, se usa la Fig. 4b. Se instancia el nodo *ControlContainer* y se definen las propiedades *HasParallelLevel* y *HasDivergentLevel*. Para diferenciar tipos de nodos de control, se incluye la propiedad *Type* (Fig. 8), con valores enumerados en la Fig. 4b.

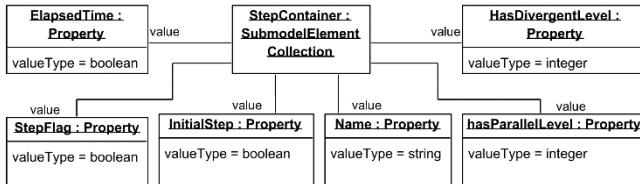


Fig. 7. Representación de *StepContainer*.

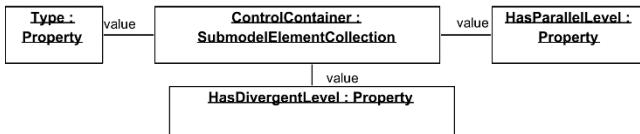


Fig. 8. Representación de *ControlContainer*.

El modelo conceptual descrito en la sección III.A permite verificar que las redes son válidas y se pueden ejecutar de manera segura y completa. Sin embargo, el metamodelo de AAS no permite modelar las reglas propuestas en este trabajo, por lo que deben considerarse durante la implementación de la herramienta de transformación basada en esta propuesta. Esta herramienta deberá verificar el cumplimiento de las restricciones OCL y, solo si se cumplen, proceder con la transformación. Como alternativa, se evaluará la viabilidad de utilizar el elemento *Operation* del metamodelo de AAS para representar las especificaciones OCL como operaciones, según lo sugerido por Olivé [22].

La Tabla I muestra un mapeo entre los conceptos del modelo propuesto para el lenguaje SFC y los del metamodelo de AAS. Las clases se mapearon a colecciones de elementos (*SubmodelElementCollection*), los atributos a propiedades (*Property*) y los enlaces a instancias de *RelationshipElement*.

C. Implementación del Submodelo

La implementación de la prueba de concepto se realizó utilizando la herramienta *AASX Package Explorer*. El caso de estudio se basa en la especificación de un procedimiento para la fabricación de pasta dental presentado en [25], y consta de tres unidades de procedimiento: *Make Original Tooth Paste*,

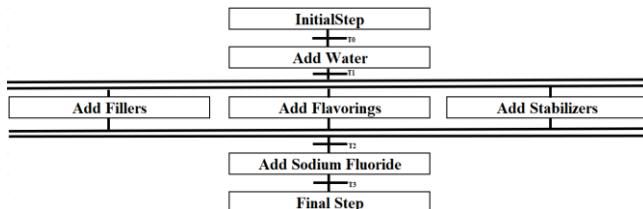


Fig. 9. Operación Agregar Ingredientes [25].

Make gel tooth paste y *Swirl the tooth paste*. En la unidad *Make gel tooth paste* se llevan a cabo las operaciones: *Add ingredients*, *React tooth paste* and *Prepare to transfer*. La operación *Add ingredients* se representa en la Fig. 9. La implementación en AASX Package Explorer se muestra en la Fig. 10, donde se ven los atributos de la red y las colecciones *StepSet*, *TransitionSet* y *ControlSet*. También se detalla la implementación del paso *Add fillers*. La serialización en XML del archivo generado está disponible en el repositorio externo¹.

TABLA I

MAPEO ENTRE CONCEPTOS DE SFC Y AAS

Concepto del lenguaje SFC	Concepto del metamodelo de AAS	Instancia del metamodelo de AAS
Elapsed Time : Property	SubmodelElementCollection	Network
StepFlag : Property	Property (boolean)	IsUnsafe
InitialStep : Property	Property (boolean)	IsUnreachable
Name : Property	Property (string)	Node
HasDivergentLevel : Property	Property (integer)	The decision was made not to map.
HasParallelLevel : Property	Property (integer)	Step
Type : Property	Property (string)	SubmodelElementCollection
HasParallelLevel : Property	Property (Date Time)	StepContainer
HasDivergentLevel : Property	Property (integer)	name
StepFlag	Property (string)	StepFlag
elapsedTime	Property (Date Time)	Elapsed Time
hasDivergentLevel	Property (integer)	HasDivergentLevel
hasParallelLevel	Property (integer)	HasParallelLevel
InitialStep	Property (boolean)	HasParallelLevel
Transition	SubmodelElementCollection	InitialStep
priority	Property (integer)	Priority
TransitionCondition	Property (string)	Priority
Control	SubmodelElementCollection	Priority
	Property (boolean)	Priority
Aggregation	Attribute value	Priority
Network - Node	RelationshipElement	Priority
Link Network - InitialStep	RelationshipElement	Step Transition
Link between nodes	RelationshipElement	Step Set
		TransitionSet
		ControlStep
		InitialStep
		StepTransition
		TransitionStep
		StepControl
		TransitionControl
		ControlStep
		ControlTransition

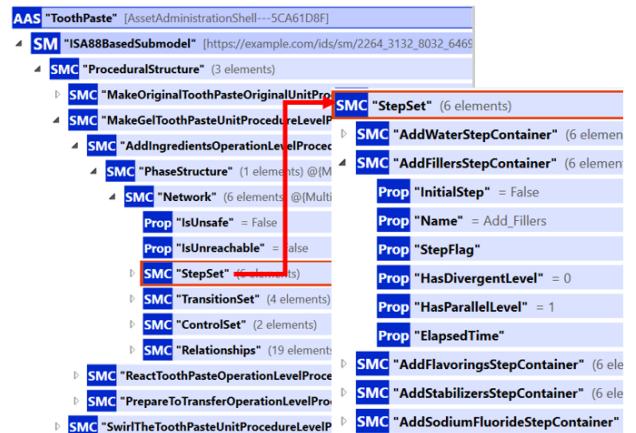


Fig. 10. Implementación del procedimiento a nivel de fase en la herramienta AASX Package Explorer.

V. DISCUSIÓN

La prueba de concepto realizada en la herramienta *AASX Package Explorer* muestra que es factible especificar procedimientos de la industria de procesos a nivel de fase

desde un modelo conceptual. Tal como se resume en la Tabla I, el concepto *Node* no fue representado como instancia del metamodelo de *AAS*, dado que este metamodelo no permite la especialización de los conceptos, no facilitando el reuso de propiedades y requiriendo que el implementador especifique en cada subclase concreta las propiedades heredadas. En consecuencia, no fue posible heredar las propiedades definidas en *Node*, teniendo que ser definidas en cada una de sus subclases. Para el caso de *Control*, se decidió no representar las subclases definidas en la Fig. 4.a, y se optó por la representación mediante atributo de la Fig. 4.b. Esto permitió la definición de una cantidad menor de conceptos en la herramienta *AASX Package Explorer*, sin embargo, no se posibilita la definición de propiedades particulares para cada tipo de nodo de control.

A pesar de que el procedimiento para la detección de redes inseguras e inalcanzables es funcional, solo puede ser usado como un paso previo a la implementación en la herramienta *AASX Package Explorer*, pudiendo cometerse errores por parte del usuario de la herramienta a la hora de la implementación. Esto se debe a que el metamodelo *AAS* no permite especificar las reglas de derivación OCL. Entonces, es necesario desarrollar una herramienta que valide automáticamente si la red es insegura o inalcanzable y lo incorpore en el submodelo de procedimientos, sin embargo, el desarrollo de dicha herramienta está fuera del alcance del presente trabajo. Por otra parte, la herramienta *AASX Package Explorer* podría ser mejorada con la inclusión de operaciones que permitan la especificación de dichas reglas, tal como lo sugiere la referencia [22].

La propuesta presentada en este trabajo permite especificar un procedimiento en sus distintos niveles. Sin embargo, dado que el lenguaje *SFC* no lo contempla, esta propuesta no posibilita representar relaciones, tales como sincronización de equipos o transferencia de materiales que pueden darse en los niveles más altos del procedimiento. A futuro se pretende poder representar el procedimiento a nivel de operación y de unidad de procedimiento, pudiendo incorporar estas interacciones propias del proceso. Para ello, el estándar ISA-88 sugiere el uso del lenguaje *Procedural Function Chart (PFC)*, el cual comparte varios de los conceptos que emplea el lenguaje *SFC*, tales como: pasos, transiciones y nodos de control, por lo que el estudio realizado en el presente trabajo puede sentar las bases para el uso de PFC.

Uno de los enfoques actuales para desarrollar gemelos digitales en la industria de procesos consiste en crear modelos computacionales de productos, procesos y recursos involucrados en la producción [26]. En este contexto, la comunicación entre máquinas se basa en el intercambio de patrones de bits entre socios de la cadena de valor [27]. El entendimiento del significado de estos patrones de bits es asegurado por una semántica implícita, donde los desarrolladores comparten un entendimiento común de los símbolos intercambiados, no obstante, en el ámbito de la Industria 4.0, donde los socios intercambian información, no se puede garantizar una interpretación fija de los símbolos [27]. El *AAS*, basado en la norma IEC 61360, permite definir propiedades y compartir atributos adicionales como

identificadores legibles por máquina y referencias a estándares, facilitando una comunicación explícita [27]. El uso de plantillas de submodelos *AAS* basadas en estándares, como se propone en este trabajo, asegura una correcta comunicación entre los actores de la cadena de valor en entornos de industria 4.0.

V. CONCLUSIONES

En este trabajo se introducen una conceptualización y su implementación para la definición de procedimientos de manufactura. Por un lado, se presenta un modelo que conceptualiza los constructores del lenguaje *SFC* y restringe la definición de redes *SFC* inválidas. Por otro lado, se especifica una instanciación del metamodelo de *AAS* propuesto por la plataforma I4.0, construido a partir del modelo conceptual del *SFC*. Por último, este trabajo contribuye con la definición de reglas de mapeo que permiten transformar del primer modelo al segundo. Esta propuesta sienta las bases para una futura automatización de la transformación entre ambos modelos, que permitiría que los expertos en la definición de redes *SFC*, no necesiten conocer los conceptos para la construcción de *AAS*, sino que puedan focalizar su actividad en la definición de procesos de manufactura utilizando un lenguaje conocido para ellos (*SFC*), delegando la generación de la representación digital de dicho proceso en una herramienta automática de transformación basada en los mapeos propuestos en este trabajo. Como trabajo futuro, se procederá a validar la propuesta con procedimientos complejos de fabricación *batch*, lo que contribuirá a consolidar su aplicabilidad. Además, se abordarán las limitaciones de la propuesta relacionadas con la imposibilidad de mapear relaciones de generalización/especialización y las reglas OCL. La construcción de prototipos que implementen las reglas de mapeo será esencial para demostrar la viabilidad de las soluciones planteadas.

REFERENCES

- [1] Z. Mueller-Zhang, P. Oliveira Antonino, and T. Kuhn, “Integrated Planning and Scheduling for Customized Production using Digital Twins and Reinforcement Learning,” *IFAC-PapersOnLine*, vol. 54, no. 1, pp. 408-413, Aug. 2021, doi: 10.1016/j.ifacol.2021.08.046.
- [2] Z. Müller-Zhang, T. Kuhn, and P. O. Antonino, “Towards live decision-making for service-based production: Integrated process planning and scheduling with Digital Twins and Deep-Q-Learning,” *Comput in Ind*, vol. 149, Jan. 2023, doi: 10.1016/j.compind.2023.103933.
- [3] A. Koulouris, N. Misailidis, and D. Petrides, “Applications of process and digital twin models for production simulation and scheduling in the manufacturing of food ingredients and products,” *Food and Bioprod. Process.*, vol. 126, pp. 317–333, Mar. 2021, doi: 10.1016/j.fbp.2021.01.016.
- [4] Bader, S., Barnstedt, E., Bedenbender, H., Berres, B., Billmann, M., & Ristin, M., “Details of the asset administration shell-part 1: the exchange of information between partners in the value chain of industrie 4.0 (version 3.0 rc02)”, Platform Industrie 4.0, Berlin, Germany, May. 2022.

- [5] J. Sheng *et al.*, "Digital twin driven intelligent manufacturing for PCB etching production line," *Comput Ind Eng*, vol. 186, Dec. 2023, doi: 10.1016/j.cie.2023.109763.
- [6] Q. Zhang *et al.*, "Digital twin-driven intelligent production line for automotive MEMS pressure sensors," *Advanced Engineering Informatics*, vol. 54, Mar. 2022, doi: 10.1016/j.aei.2022.101779.
- [7] P. Sapel *et al.*, "Towards digital shadows for production planning and control in injection molding," *CIRP J Manuf Sci Technol*, vol. 38, pp. 243–251, Aug. 2022, doi: 10.1016/j.cirpj.2022.05.003.
- [8] C. Zhang, W. Xu, J. Liu, Z. Liu, Z. Zhou, and D. T. Pham, "A Reconfigurable Modeling Approach for Digital Twin-based Manufacturing System," *Procedia CIRP*, vol. 83, pp. 118–125, 2019, doi: 10.1016/j.procir.2019.03.141.
- [9] *Batch Control, Part 1: Models and Terminology*, ISA-88.01-1995, 2006.
- [10] *Batch Control Part 2: Data Structures and Guidelines for Languages*, ANSI/ISA-88.00.02, 2001.
- [11] R. Pairol-Fuentes, J. Alvarado-Domínguez, M. Vegetti, L. Roldán, and S. Gonnet, "Modelo para la digitalización de activos de la industria batch basado en el estándar ISA 88," *Memorias De Las JAIIo*, vol. 8, no. 13, pp. 33–46, 2022. [Online]. Available: <https://publicaciones.sadio.org.ar/index.php/JAIIo/article/view/316/263>.
- [12] D. C. Schmidt, "Model-Driven Engineering," 2006. [Online]. Available: <http://mic.omg.org>
- [13] *Reference Architecture Model Industrie 4.0 (RAMI 4.0)*, DIN SPEC 91345, 2016.
- [14] *Programmable controllers - Part 3: Programming languages*, IEC 61131-3, 2013.
- [15] J. H. Hang, W. P. Lee and Y. M. Lim, "Towards Mass Individualized Production: RAMI 4.0 Asset Data Channelling for Manufacturing Value Chain Connectivity," *2021 3rd InterN. Conf. on Comp. Commun. and the Internet*, Nagoya, Japan, 2021, pp. 225-231, doi: 10.1109/ICCCI51764.2021.9486787.
- [16] S. J. Shin and J. Um, "Deploying data analytics models in asset administration shells: Energy prediction in manufacturing," *Eng. Appl. Artif. Intell.*, vol. 138, Dec. 2024, doi: 10.1016/j.engappai.2024.109269.
- [17] M. Damm, T. Treichel, A. L. T. Costa, P. Niemann-Stryczek, M. Esser and A. Hibbeln, "Tracking the power consumption of concurrent batches in production processes," *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sinaia, Romania, 2023, pp. 1-7, doi: 10.1109/ETFA54631.2023.10275658.
- [18] W. Motsch, A. Sidorenko, A. David, P. Rübel, A. Wagner, and M. Ruskowski, "Electrical energy consumption interface in modular skill-based production systems with the asset administration shell," in *Procedia Manufacturing*, vol 55, pp. 535–542. Oct. 2021. doi: 10.1016/j.promfg.2021.10.073.
- [19] B. Lv, Y. Zhang and W. Dai, "Integrated Asset Management Model based on IEC 61499 and Administration Shell," *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, Toronto, ON, Canada, 2021, pp. 1-6, doi: 10.1109/IECON48115.2021.9589597.
- [20] S. Cavalieri and M. G. Salafia, "Asset Administration Shell for PLC Representation Based on IEC 61131-3," in *IEEE Access*, vol. 8, pp. 142606-142621, 2020, doi: 10.1109/ACCESS.2020.3013890.
- [21] S. Grüner, M. Hoernicke, K. Stark, N. Schoch, N. Eskandani, and J. Pretlove, "Towards asset administration shell-based continuous engineering in process industries," *At-Automatisierungstechnik*, vol. 71, no. 8, pp. 689–708, Aug. 2023, doi: 10.1515/auto-2023-0012.
- [22] Antoni Olivé, *Conceptual Modeling of Information Systems*. Berlin, Heidelberg: Springer-Verlag, 2007.
- [23] K. H. John and M. Tiegelkamp, "The Programming Languages of IEC 61131-3," in *IEC 61131-3: Programming Industrial Automation Systems*, 2nd ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 99-205, doi: 10.1007/978-3-642-12015-2.
- [24] 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.
- [25] J. Parshall and L. Lamb, *Applying S88: Batch Control from a User's Perspective*. 1999.
- [26] P. Verboven, T. Defraeye, A. K. Datta, and B. Nicolai, "Digital twins of food process operations: the next step for food process models?," *Curr Opin Food Sci*, vol. 35, pp. 79–87, Oct. 2020, doi: 10.1016/j.cofs.2020.03.002.
- [27] *Language for 4.0 components*, VDI/VDE 2193 part 1, 2020.



Johnny Alvarado Domínguez received the B.S. in industrial engineering from Universidad Simon Bolivar, Barranquilla, Colombia. He is currently pursuing a Ph.D. degree in industrial engineering at Universidad Tecnológica Nacional, Santa Fe, Argentina. Since 2021 he has been a doctoral scholar at the INGAR, Santa Fe, Argentina. He is currently a researcher at Aimen Technology Center, Spain. His research interests include Industry 4.0 and Ontologies.



Marcela Vegetti received her PhD in Information Systems Engineering from Universidad Tecnológica Nacional in 2007. She is a professor in the Department of Information Systems Engineering at the same university and holds a research position at CONICET. Her research interests include applying ontologies to conceptual modeling.



Silvio Gonnet received his PhD degree in Engineering from Universidad Nacional del Litoral in 2003. He is a professor at the Department of Information Systems Engineering of the Facultad Regional Santa Fe, Universidad Tecnológica Nacional (Santa Fe, Argentina). He also holds a Researcher position at CONICET. His research interests are conceptual modeling, and discrete-event modeling & simulation.