

Design of a UML Profile for the Conceptual Modeling of Cyber-Physical Systems

Hiembuchner, Florencia Abigail.

Universidad Tecnológica Nacional, Santa Fe, Argentina.
florhiembuchner@gmail.com

Abstract. This paper presents a proposal for modeling Cyber-Physical Systems (CPS) through the definition of a specialized UML (Unified Modeling Language) profile. The proposal extends the UML component diagram to explicitly capture the structural characteristics inherent to CPS. The proposed profile enables a clear and consistent representation of the different components that constitute such systems, encompassing both physical elements and components belonging to the computational and communication domains. Additionally, the profile supports the development of models by enabling the integration of OCL (Object Constraint Language) rules that ensure the structural consistency of the final models. The main objective of this proposal is to provide a clear and low-complexity modeling mechanism, to improve the development of consistent CPS models within UML-based design environments.

Keywords: design of CPS, structural model, UML component diagram, OCL constraints.

Diseño de un Perfil UML para el Modelado Conceptual de Sistemas Ciberfísicos

Resumen. Este artículo presenta una propuesta para modelar Sistemas Ciberfísicos (CPS) mediante la definición de un perfil especializado de UML (Lenguaje Unificado de Modelado). La propuesta extiende el diagrama de componentes de UML para capturar explícitamente las características estructurales inherentes a los CPS. El perfil propuesto permite una representación clara y consistente de los diferentes componentes que constituyen estos sistemas, abarcando tanto elementos físicos como componentes pertenecientes a los dominios computacional y de comunicación. Además, el perfil facilita el desarrollo de modelos al permitir la integración de reglas OCL (Lenguaje de Restricciones de Objetos), las cuales garantizan la consistencia estructural de los modelos finales. El objetivo principal de esta propuesta es proporcionar un mecanismo de modelado claro y de baja complejidad, con el fin de mejorar el desarrollo de modelos consistentes de CPS dentro de entornos de diseño basados en UML.

Palabras clave: diseño de CPS, modelo estructural, diagrama de componentes UML, restricciones OCL.

1 Introducción

En la actualidad, los sistemas ciberfísicos han adquirido una presencia creciente en diversos ámbitos de la vida cotidiana y en múltiples sectores industriales. Sin embargo, aún persisten limitaciones en cuanto a las herramientas y enfoques disponibles para su adecuada representación. Un sistema ciberfísico (CPS) puede definirse como una integración estrecha entre componentes físicos y computacionales, donde ambos dominios interactúan de manera continua y en tiempo real (Edward A. Lee, 2008). Estos sistemas utilizan sensores para la adquisición de información del entorno físico y actuadores para intervenir sobre él, lo que permite implementar mecanismos de monitoreo, control y retroalimentación. Como resultado, los CPSs posibilitan niveles avanzados de automatización, facilitan la toma de decisiones autónoma y contribuyen a la optimización del rendimiento de los procesos en los que se aplican.

El modelado de este tipo de sistemas constituye un desafío significativo, ya que los modelos deben representar de manera precisa la estructura y los múltiples componentes que lo conforman. La construcción de estos modelos no solo requiere conocimientos específicos sobre enfoques de modelado, sino también experiencia en el dominio de aplicación. Luego, es necesario contar con herramientas que faciliten la elaboración de representaciones claras, consistentes y adecuadas para el posterior análisis del sistema.

En este contexto, UML (Unified Modeling Language), estandarizado por el Object Management Group (OMG), (Object Management Group, 2017), constituye uno de los lenguajes de modelado más utilizados para la especificación de sistemas de software. No obstante, cuando UML se aplica directamente a dominios específicos, surgen limitaciones que dificultan su utilización. En particular, la falta de especialización del lenguaje puede conducir a interpretaciones heterogéneas entre modeladores, dado que los modelos suelen reflejar la perspectiva individual de quien los construye más que una representación semiformal y consensuada del sistema bajo análisis.

Con el objetivo de abordar estas limitaciones, este trabajo propone el desarrollo de un perfil UML orientado al modelado conceptual de sistemas ciberfísicos. De acuerdo con Eriksson et al. (2004), un perfil UML constituye un mecanismo de extensión del lenguaje que permite adaptar su metamodelo a dominios específicos mediante estereotipos, valores etiquetados y restricciones. Este enfoque permite aprovechar un estándar ampliamente adoptado, evitando la necesidad de definir un nuevo lenguaje de modelado y manteniendo la compatibilidad con herramientas de modelado existentes.

El resto del artículo se organiza de la siguiente manera. En la Sección 2 se presenta el marco teórico. En la Sección 3 se describe el perfil UML desarrollado. La Sección 4 presenta la validación del perfil mediante un caso de estudio. La Sección 5 describe el análisis y discusión de los resultados obtenidos, mientras que la Sección 6 presenta las conclusiones del trabajo junto con posibles líneas de trabajo futuro.

2 Marco Teórico

2.1 Fundamentos de los Sistemas Ciberfísicos

Un CPS puede definirse como la integración de infraestructura computacional y componentes físicos que interactúan mediante mecanismos de control y comunicación.

En este tipo de sistemas, el mundo físico y el mundo digital se encuentran estrechamente interconectados, permitiendo el monitoreo continuo del entorno y la ejecución de acciones sobre el mismo (Figura 1).

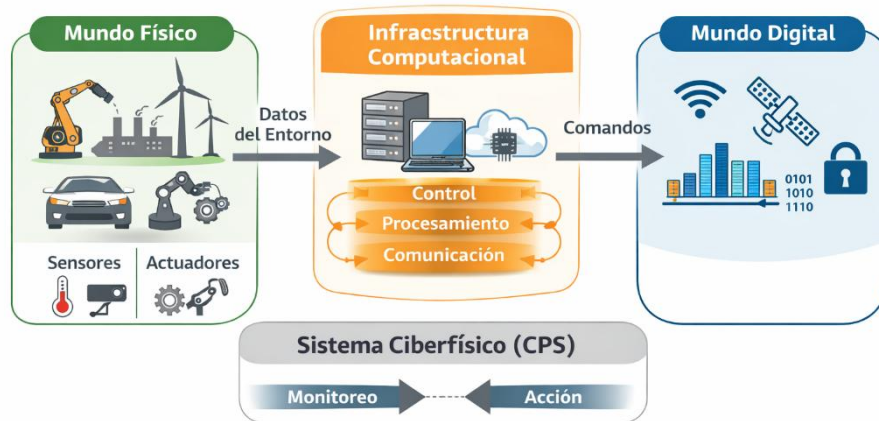


Fig. 1. Esquema de un CPS que ilustra la interacción entre el mundo físico y la infraestructura computacional, así como los procesos de monitoreo y acción.

De acuerdo con Cardin (2019), la interacción entre el mundo físico y el ciberespacio puede describirse a partir de la identificación de tres tipos de componentes fundamentales: *control*, que permite al CPS interactuar con el mundo físico, capturando información mediante sensores y actuando sobre dispositivos físicos; *computación*, encargada de transformar los datos provenientes del mundo físico en información útil para la toma de decisiones; y *comunicación*, que posibilita el intercambio de información entre los distintos componentes del sistema.

El perfil UML propuesto en la Sección 3 se fundamenta en dicha clasificación, la cual se adopta como criterio para la definición de los estereotipos y las relaciones del modelo, con el propósito de representar de manera consistente las características inherentes a los CPS.

2.2 Diagrama de Componentes UML

El diagrama de componentes de UML, (Object Management Group, 2017), es un tipo de diagrama estructural que permite representar la arquitectura de un sistema mediante la identificación de sus componentes y las relaciones de dependencia e interacción entre ellos. En este contexto, un componente se entiende como una unidad modular que encapsula funcionalidad y expone interfaces bien definidas, facilitando así la abstracción y el análisis de sistemas complejos.

A diferencia de otros diagramas estructurales, como el diagrama de clases (que se orienta a describir la estructura interna y los detalles de implementación), el diagrama de componentes proporciona una visión de más alto nivel, centrada en la organización del sistema y en la forma en que sus partes se conectan e interactúan. Esta característica resulta particularmente relevante en dominios donde coexisten múltiples tipos de entidades.

En este sentido, la elección del diagrama de componentes como base para la definición del perfil UML se fundamenta en su capacidad para modelar de manera adecuada la estructura de CPS, los cuales integran elementos físicos y computacionales que interactúan entre sí. La naturaleza distribuida de estos sistemas requiere una representación que permita capturar tanto los componentes como sus interfaces y relaciones, aspecto que no es abordado con igual nivel de adecuación por otros tipos de diagramas.

En este marco, se propone la definición de un perfil UML específico orientado a facilitar el modelado de CPS. La propuesta se basa en la extensión del diagrama de componentes, aprovechando su capacidad para describir de manera integrada tanto la estructura del sistema como las interacciones entre sus elementos. En particular, la incorporación de estereotipos, valores etiquetados y restricciones permite adaptar la notación a las características propias de los CPS, enriqueciendo su capacidad expresiva sin perder coherencia con el estándar UML.

3 Perfil UML: Diseño e Implementación

3.1 Diseño

El perfil propuesto busca representar de forma explícita los componentes fundamentales que intervienen en un sistema ciberfísico, permitiendo diferenciar los elementos correspondientes al mundo físico y aquellos pertenecientes al dominio computacional y de comunicación.

A partir del análisis del dominio, se identificaron los principales conceptos que deben ser representados dentro del modelo y se definieron los estereotipos que conforman el perfil. Los estereotipos permiten clasificar los elementos de un modelo y proporcionarles significados específicos del dominio de la simulación (Object Management Group, 2017). En la Tabla 1 se presentan los distintos estereotipos definidos y las metaclases a las que extienden, mientras que en la Tabla 2 se muestra la jerarquía establecida entre dichos estereotipos. Estos estereotipos permiten capturar la estructura del sistema, describiendo tanto los componentes involucrados como las relaciones existentes entre ellos.

Los valores etiquetados se utilizan para almacenar información adicional sobre los elementos del modelo. Para la definición del perfil, se utilizaron los valores detallados en la Tabla 3, acompañados también de su tipo, relación con el estereotipo correspondiente y una breve descripción de su uso. Estos valores enriquecen la representación del modelo, proporcionando mayor detalle sobre la naturaleza de las interacciones entre componentes.

Tabla 1. Estereotipos definidos para el perfil CPS y metaclases extendidas.

Estereotipo	Metaclase	Descripción
CPSModel	Model	Representa el modelo principal de un CPS, dentro del cual se definen y organizan los distintos elementos del perfil CPS.
CPSInterface	Interface	Representa una interfaz utilizada para definir los puntos de interacción y comunicación entre componentes del CPS.

CPSPort	Component	Representa un puerto asociado a un componente del sistema, a través del cual se realiza la comunicación o el intercambio de información con otros elementos
CPSComponent	Component	Representa un componente del CPS, pudiendo corresponder tanto a elementos físicos como computacionales.
CPSDependency	Dependency	Representa una relación de dependencia entre elementos del sistema, utilizada para modelar interacciones o requerimientos entre componentes e interfaces.

Tabla 2. Estereotipos definidos para el perfil CPS y su relación de especialización.

Estereotipo	Supertipo	Descripción
HumanMachine Interface	CPSInterface	Representa una interfaz destinada a la interacción entre el CPS y los usuarios humanos, permitiendo la visualización de información o el ingreso de comandos.
Signal	CPSInterface	Modela una interfaz utilizada para la transmisión de señales entre componentes del sistema.
PhysicalStimulus	CPSInterface	Representa una interfaz a través de la cual un componente recibe estímulos del entorno físico.
PhysicalAction	CPSInterface	Modela una interfaz mediante la cual un componente produce acciones o efectos sobre el entorno físico.
AnalogPort	CPSPort	Representa un puerto destinado a la transmisión o recepción de señales analógicas entre componentes.
DigitalPort	CPSPort	Representa un puerto destinado a la transmisión o recepción de señales digitales entre componentes.
Provides	CPSDependency	Indica que un componente provee o expone una interfaz o servicio a otros componentes del sistema.
Requires	CPSDependency	Indica que un componente requiere o consume una interfaz o servicio provisto por otro componente.
Delegates Communication Through	CPSDependency	Representa una dependencia mediante la cual la comunicación entre componentes es delegada a través de otro elemento del sistema.
Sensor	CPSComponent	Representa un componente encargado de percibir información del entorno físico y transformarla en datos utilizables por el sistema.
Actuator	CPSComponent	Representa un componente que ejecuta acciones sobre el entorno físico en respuesta a comandos generados por el sistema.
Computation	CPSComponent	Representa un componente responsable del procesamiento de información y la ejecución de la lógica computacional del sistema.
Physical Component	CPSComponent	Representa un componente perteneciente al dominio físico que interactúa directamente con el entorno o forma parte de la infraestructura física del sistema.

Tabla 3. Valores etiquetados, tipo, estereotipo y descripción.

Nombre	Tipo	Estereotipo Asociado	Descripción
Type	InOutType	Requires	Indica el tipo de flujo de información o señal que el componente necesita recibir a través del puerto para poder operar.

Type	InOutType	Provides	Especifica el tipo de flujo de información o señal que el componente pone a disposición de otros componentes del sistema.
Type	InOutType	Delegates Communication Through	Define el tipo de flujo de información o señal que se delega en el puerto a fin comunicar componentes internos y externos.
Type	InOutType	PhysicalAction	Indica el tipo de acción física que el componente puede ejercer sobre el entorno físico del sistema.
Type	InOutType	PhysicalStimulus	Representa el tipo de estímulo físico proveniente del entorno que puede ser detectado o recibido por el sistema.
Type	InOutType	Signal	Especifica el tipo de señal que se transmite entre componentes dentro del CPS.
Type	InOutType	HumanMachine Interface	Define el tipo de intercambio de información que se produce entre el sistema y el usuario a través de una interfaz humano-máquina.
Values	String	AnalogicPort	Especifica los valores o rangos de valores analógicos que pueden ser transmitidos o recibidos a través del puerto analógico.
values	String	DigitalPort	Define los posibles valores discretos o estados lógicos que pueden ser transmitidos o recibidos mediante el puerto digital.
objective	String	CPSModel	Describe el objetivo o propósito principal del modelo de CPS representado.

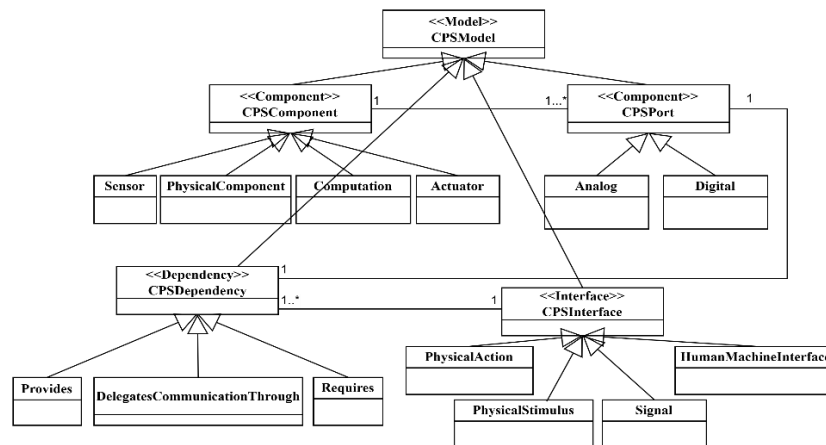


Fig. 2. Esquema del perfil UML diseñado.

El perfil resultante integra todos los elementos definidos según las Tablas 1 a 3. La Figura 2 esquematiza el diseño final haciendo uso de un diagrama de clases UML. Este diagrama resume la estructura propuesta y permite visualizar cómo los estereotipos se vinculan con las metACLases correspondientes, así como las propiedades que caracterizan a cada elemento.

El conjunto de elementos previamente detallados se complementó con invariantes expresadas en Object Constraint Language (OCL), (Object Management Group, 2014) para garantizar la validez de los modelos que apliquen el perfil UML. Las restricciones documentadas en la Tabla 4 permiten establecer reglas sobre los atributos, relaciones e interacciones de los componentes, evitando errores conceptuales y asegurando la construcción de modelos bien conformados.

Tabla 4. Descripción de las restricciones aplicadas al modelo.

Nombre	Contexto	Expresión OCL
OnlyStereotyped	CPSModel	self.base_Model.packagedElement->forAll(e e.getAppliedStereotypes()->exists(s s.oclIsKindOf(cpsprofile::CPSComponent) or s.oclIsKindOf(cpsprofile::CPSInterface) or s.oclIsKindOf(cpsprofile::CPSPort) or s.oclIsKindOf(cpsprofile::CPSDependency))))
CPSComponentNot Isolated	CPSComponent	self.base_Component.association->exists(a a.memberEnd->exists(e e.type.getAppliedStereotypes()->exists(s s.oclIsKindOf(cpsprofile::CPSPort))))
CPSInterfaceNot Isolated	CPSInterface	self.base_Interface.association->exists(a a.memberEnd->exists(end end.type <> self.base_Component and end.type.getAppliedStereotypes()->exists(s s.oclIsKindOf(cpsprofile::CPSPort))))
ProviderFrom CPSPortToCPS Interface	Provides	self.base_Dependency.client->forAll(c c.oclIsKindOf(Port) and c.getAppliedStereotype('CPSPort') <> null) and self.base_Dependency.supplier->forAll(s s.oclIsKindOf(Interface) and s.getAppliedStereotype('CPSInterface') <> null)
RequirerFrom CPSPortToCPS Interface	Requires	self.base_Dependency.client->forAll(c c.oclIsKindOf(Port) and c.getAppliedStereotype('CPSPort') <> null) and self.base_Dependency.supplier->forAll(s s.oclIsKindOf(Interface) and s.getAppliedStereotype('CPSInterface') <> null)
NoSelfRequired Interface	CPSComponent	not self.base_Component.clientDependency->exists(d1, d2 d1.getAppliedStereotype('Requires') <> null and d2.getAppliedStereotype('Provides') <> null and d1.supplier = d2.supplier)
CPSModelMustContain	CPSModel	self.base_Model.packagedElement->exists(e e.getAppliedStereotype('cpsprofile::Computati on') <> null) and self.base_Model.packagedElement->exists(c

		e.getAppliedStereotype('cpsprofile::Sensor') <> null) and self.base_Model.packagedElement->exists(e e.getAppliedStereotype('cpsprofile::PhysicalComponent') <> null)
CPSInterface SingleProvider	CPSInterface	self.base_Interface ->select(d d.getAppliedStereotype('Provides') <> null) ->size() <= 1

3.2 Implementación

Para la implementación del perfil UML propuesto se utilizó la herramienta Eclipse Papyrus (Eclipse Foundation, s. f.), un entorno de modelado orientado al Desarrollo Dirigido por Modelos (Schmidt, 2006) que permite trabajar con estándares como UML y SYSML (Systems Modeling Language) (Object Management Group, 2024). Papyrus proporciona soporte para la definición y aplicación de perfiles UML, permitiendo extender el metamodelo mediante estereotipos, valores etiquetados y restricciones. Esta característica resulta especialmente adecuada para el desarrollo del perfil propuesto, ya que posibilita incorporar conceptos específicos del dominio de los sistemas ciberfísicos dentro de un entorno de modelado estandarizado.

El uso de una herramienta de modelado compatible con estándares facilita la reutilización del perfil en distintos entornos de desarrollo y permite validar automáticamente las restricciones definidas sobre los modelos construidos. Asimismo, la integración entre UML, OCL y el entorno de modelado permite mejorar la calidad de los modelos generados, reduciendo inconsistencias y proporcionando mecanismos de verificación temprana durante el proceso de diseño. Por otro lado, al formar parte del ecosistema de Eclipse, la herramienta facilita su integración con otras herramientas de la plataforma y posibilita su extensión hacia nuevas áreas o funcionalidades como parte de trabajos futuros.

La validación del perfil implementado se realizó mediante un proceso iterativo. A medida que modelaron casos de estudio simples, se identificaron nuevas necesidades de representación y aspectos del dominio que no habían sido considerados inicialmente. Este proceso permitió refinar progresivamente el perfil mediante la incorporación de nuevos estereotipos, ajustes en las relaciones entre elementos y la definición de restricciones adicionales que garantizan la consistencia del modelo.

4 Caso de Estudio: Sistema Industrial de Brazo Robótico

Con el propósito de evaluar la utilidad y aplicabilidad del perfil UML propuesto, se emplea como caso de estudio un CPS correspondiente a un sistema industrial de brazo robótico. Este CPS integra componentes físicos y computacionales que interactúan para detectar objetos, controlar el movimiento del brazo robótico y ejecutar acciones sobre el entorno. El sistema incluye diversos componentes tales como sensores de posición, detectores de obstáculos, controladores del brazo robótico y de la pinza robótica, así como una interfaz de usuario que permite la interacción con el operador. Los sensores captan información del entorno físico y la transmiten al controlador principal del

sistema, mientras que los actuadores ejecutan las acciones físicas correspondientes a partir de las decisiones generadas por el sistema computacional.

Durante el proceso de modelado se aplicaron los estereotipos definidos en el perfil para representar los distintos elementos del sistema y las relaciones entre ellos. Las Tablas 5, 6, 7 y 8 presentan la especificación de los estereotipos requeridos para cada uno de los elementos del caso de estudio, incluyendo componentes, puertos, interfaces y dependencias. Por otra parte, la Figura 3 corresponden al modelo construido en la herramienta de modelado Papyrus aplicando el perfil definido según estas especificaciones.

Como puede observarse, la aplicación del perfil permitió modelar de manera clara las relaciones entre los distintos elementos del sistema, en tanto estas se representan explícitamente mediante conectores e interfaces, facilitando la identificación de dependencias y flujos de comunicación. En este sentido, si bien las Tablas 5 a 8 contienen información equivalente, su presentación en forma tabular no permite visualizar de manera inmediata las relaciones entre los elementos. Por el contrario, la representación gráfica mediante el diagrama favorece una comprensión más intuitiva de la arquitectura general del CPS, al integrar en una única vista tanto los componentes como sus conexiones. Asimismo, las restricciones definidas en el perfil contribuyen a verificar la consistencia estructural del modelo generado.

Tabla 5. Aplicación del estereotipo CPSComponent en el caso práctico.

Nombre	Estereotipo	Descripción
Position Sensor	Sensor	Sensor encargado de detectar la posición de los objetos en el área de trabajo del brazo robótico, determinando si estos se encuentran dentro o fuera del alcance del manipulador.
Environment	Physical Component	Representa el entorno físico en el cual opera el sistema, incluyendo los objetos presentes y las condiciones que pueden influir en el funcionamiento del brazo robótico.
Obstacle Detector	Sensor	Sensor destinado a identificar la presencia de obstáculos en el entorno de operación, permitiendo prevenir colisiones durante el movimiento del brazo robótico.
RobotArm Controller	Computation	Componente computacional responsable de procesar la información proveniente de los sensores y de coordinar las acciones del sistema, generando las órdenes necesarias para el actuador.
User Interface	Computation	Componente que permite la interacción entre el usuario y el sistema, facilitando el envío de comandos y la visualización del estado del proceso.
Gripper Controller	Actuator	Actuador encargado de ejecutar acciones físicas sobre el entorno, específicamente abrir o cerrar la pinza del brazo robótico para manipular objetos.

Tabla 6. Aplicación del estereotipo CPSPort en el caso práctico.

Nombre	Estereotipo	Descripción
within reach-out of reach sensor	DigitalPort	Puerto digital que transmite la información generada por el sensor de posición indicando si el objeto se encuentra dentro o fuera del alcance del brazo robótico.

within reach-out of reach environment	DigitalPort	Puerto digital que representa la información del entorno físico respecto a la ubicación del objeto en relación con el alcance del manipulador.
obstacle-no obstacle environment	DigitalPort	Puerto digital que envía información sobre la presencia o ausencia de obstáculos detectados en el entorno físico.
obstacle-no obstacle detector	DigitalPort	Puerto digital mediante el cual el sensor de obstáculos envía la información sobre la detección de obstáculos al sistema de control.
start-stop-reset user	DigitalPort	Puerto digital utilizado por la interfaz de usuario para enviar comandos de inicio, detención o reinicio del sistema.
within reach-out of reach controller	DigitalPort	Puerto digital del controlador que recibe información sobre la posición del objeto en relación con el alcance del brazo.
obstacle-no obstacle controller	DigitalPort	Puerto digital del controlador que recibe información sobre la presencia o ausencia de obstáculos en el entorno.
start-stop-reset controller	DigitalPort	Puerto digital del controlador que recibe los comandos enviados por el usuario para iniciar, detener o reiniciar el sistema.
open-close status requires	DigitalPort	Puerto digital a través del cual el controlador solicita información sobre el estado actual de la pinza del actuador.
success-error user	DigitalPort	Puerto digital que transmite a la interfaz de usuario información sobre el resultado de las operaciones realizadas por el sistema.
open-close status provides	DigitalPort	Puerto digital del actuador que proporciona información acerca del estado de apertura o cierre de la pinza.
open-close command provides	DigitalPort	Puerto digital del controlador mediante el cual se envían comandos al actuador para abrir o cerrar la pinza.
success-error controller	DigitalPort	Puerto digital que comunica al controlador el resultado de la ejecución de una acción realizada por el actuador.
open-close command requires	DigitalPort	Puerto digital del actuador que recibe las órdenes provenientes del controlador para abrir o cerrar la pinza.

Tabla 7. Aplicación del estereotipo CPSInterface en el caso práctico.

Nombre	Estereotipo	Descripción
ItemPosition	Physical Stimulus	Interfaz que representa la información física relacionada con la posición de un objeto dentro del entorno del sistema.
Obstacle Detector	Physical Stimulus	Interfaz que describe la información sobre la detección de obstáculos en el entorno físico del sistema.
User Commands	HumanMachine Interface	Interfaz que define los comandos que el usuario puede enviar al sistema para controlar su funcionamiento.
Gripper Status	Signal	Interfaz que comunica el estado actual del actuador, indicando si la pinza del brazo robótico se encuentra abierta o cerrada.
Gripper Command	PhysicalAction	Interfaz que define las acciones físicas que el sistema puede ejecutar sobre el actuador, como abrir o cerrar la pinza.
Feedback ToUser	HumanMachine Interface	Interfaz que permite enviar retroalimentación al usuario sobre el estado o resultado de operaciones realizadas por el sistema.

Tabla 8. Aplicación del estereotipo CPSDependency en el caso práctico.

Componente inicio	Componente	Estereotipo específico
-------------------	------------	------------------------

	Fin	
PositionSensor	within reach-out of reach sensor	DelegatesCommunicationThrough
Environment	within reach-out of reach environment	DelegatesCommunicationThrough
Environment	obstacle-noobstacle environment	DelegatesCommunicationThrough
ObstacleSensor	obstacle-noobstacle detector	DelegatesCommunicationThrough
within reach-out of reach sensor	ItemPosition	Provides
within reach-out of reach environment	ItemPositiion	Requires
obstacle-no obstacle environment	ObstacleDetector	Requires
obstacle-no obstacle detector	ObstacleDetector	Provides
UserInterface	start-stop-reset user	DelegatesCommunicationThrough
start-stop-reset user	UserCommands	Requires
start-stop-reset controller	UserCommands	Requires
obstacle-no obstacle controller	ObstacleDetector	Requires
RobotArmController	start-stop-reset controller	DelegatesCommunicationThrough
RobotArmController	obstacle-noobstacle controller	DelegatesCommunicationThrough
RoborArmController	within reach-out of reach controller	DelegatesCommunicationThrough
RobotArmController	open-closestatus requires	DelegatesCommunicationThrough
RobotArmController	open-closecommand provides	DelegatesCommunicationThrough
RoborArmController	success-error user	DelegatesCommunicationThrough
UserInterface	Success-error user	DelegatesCommunicationThrough
success-error user	FeedbackToUser	Provides
success-error controller	FeedbackToUser	Provides
open-closestatus requires	GripperStatus	Requires
open-closestatus provides	GripperStatus	Provides
open-close command provides	GripperCommand	Provides
open-close command requires	GrippeCommand	Requires
GripperController	open-close status provides	DelegatesCommunicationThrough
GripperController	open-close command requires	DelegatesCommunicationThrough

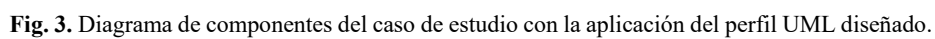


Fig. 3. Diagrama de componentes del caso de estudio con la aplicación del perfil UML diseñado.

Los resultados obtenidos evidencian que el perfil UML propuesto constituye una herramienta adecuada para representar la estructura de sistemas ciberfísicos, permitiendo construir modelos haciendo uso de un entorno de modelado estandarizado. En particular, el caso de estudio demuestra que el perfil facilita la rápida identificación de componentes físicos, elementos computacionales y mecanismos de comunicación, aspectos fundamentales en el modelado de este tipo de sistemas.

5 Discusión

La propuesta presentada en este trabajo tiene como objetivo proporcionar un mecanismo estructurado para el modelado conceptual de sistemas ciberfísicos. Uno de los principales aportes del perfil UML desarrollado es la posibilidad de expresar explícitamente los elementos característicos de este tipo de sistemas, permitiendo diferenciar claramente entre componentes físicos, computacionales y de comunicación. Esta diferenciación resulta fundamental para representar adecuadamente la interacción entre el mundo físico y el mundo digital que caracteriza a este tipo de sistemas.

El uso de un perfil UML presenta además diversas ventajas. En primer lugar, permite aprovechar un estándar consolidado, evitando la necesidad de desarrollar un nuevo lenguaje de modelado desde cero. En segundo lugar, facilita la utilización de herramientas existentes, lo que reduce la complejidad asociada a la adopción de la propuesta.

La incorporación de restricciones mediante OCL constituye otro aspecto relevante del perfil, ya que permite validar automáticamente determinadas reglas del dominio. De esta forma, es posible detectar inconsistencias estructurales en los modelos durante las primeras etapas del proceso de diseño, contribuyendo a mejorar la calidad de las representaciones generadas.

La aplicación del perfil en el caso de estudio permitió comprobar que los estereotipos definidos resultan adecuados para representar los componentes principales de un CPS, así como las relaciones que se establecen entre ellos. El modelo obtenido permitió visualizar de manera clara la arquitectura general del sistema y comprender cómo se produce la interacción entre los distintos elementos.

6 Conclusiones y Trabajos Futuros

En este trabajo se presenta una propuesta para el modelado de CPSs basada en la definición de un perfil UML especializado. Este enfoque consiste en extender el diagrama de componentes con el fin de adaptarlo a las características propias de este tipo de sistemas.

La implementación del perfil utilizando la herramienta Papyrus permitió construir modelos consistentes y verificar automáticamente el cumplimiento de reglas estructurales definidas como invariantes OCL. Asimismo, la aplicación del perfil en un caso de estudio demuestra la viabilidad de la propuesta junto con su potencial de representación en comparación con un listado tabulado.

Como líneas de trabajo futuro, se propone ampliar el perfil para representar comportamientos dinámicos del sistema e integrarlo con otras herramientas de

modelado. También, se plantea desarrollar un mecanismo de exportación automática de los modelos a formatos estructurados que permitan su utilización en entornos de simulación con objetivos de estudio de comportamiento y análisis.

Referencias

- Alur, R. (2015). *Principles of Cyber-Physical Systems*. Cambridge, MA: MIT Press. Berkeley.
- Cardin, O. (2019). Classification of cyber-physical production systems applications: Proposition of an analysis framework. *Computers in Industry*, 104, 11-21.
- Douglas C. Schmidt. (2006). *Model-driven engineering*. IEEE Computer, 39(2), 25–31.
- Eclipse Foundation. (s. f.). *Eclipse Papyrus*™. <https://eclipse.dev/papyrus/>
- Eriksson, H.-E., Penker, M., Lyons, B., & Fado, D. (2004). *UML 2 toolkit*. Wiley Publishing.
- Lee, E. A. (2008). *Cyber Physical Systems: Design Challenges*. University of California, Object Management Group. (2014). Object Constraint Language (OCL), version 2.4. <https://www.omg.org/spec/OCL>
- Object Management Group. (2017). Unified Modeling Language (UML), version 2.5.1. <https://www.omg.org/spec/UML>
- Object Management Group. (2024). *SysML® v1 specification*. <https://www.omg.org/sysml/sysmlv1/>
- Robinson, S. (2014). *The practice of model development and use* (2nd ed.). New York, NY: Palgrave Macmillan.