# Ontología para la Representación de Entidades con Comportamientos basados en Eventos

# A Core Ontology for the Representation of Entities with Event-based Behaviors

María Julia Blas[1], Silvio Gonnet[1], Pablo Becker[2], Luis Olsina[2]

[1] Instituto de Desarrollo y Diseño INGAR, CONICET-UTN, Santa Fe, Argentina
{mariajuliablas,sgonnet}@santafe-conicet.gov.ar
[2] GIDIS_Web, Facultad de Ingeniería, UNLPam, General Pico, LP, Argentina
{beckerp,olsinal}@ing.unlpam.edu.ar

**Abstract.** El modelado de comportamientos dinámicos usando eventos como disparadores del cambio de estado de entidades es un área de interés en Modelado y Simulación (M&S). En este trabajo se presentan los términos, propiedades, relaciones y axiomas de ParticularEventCO (PEventCO) como un modelo semántico basado en la noción de eventos causados por el comportamiento de entidades. Esta ontología se encuentra situada en el nivel "Core" de la arquitectura FCD-OntoArch, donde la ontología ThingFO define el nivel fundacional. El principal objetivo es lograr una representación del comportamiento de entidades dinámicas siguiendo un enfoque basado en eventos como complemento de las ontologías ya existentes en FCD-OntoArch. Luego, se detalla la forma en la cual los principales elementos del nivel fundacional han sido redefinidos en PEventCO. Además, se incluye la especificación formal del modelo haciendo uso de ConceptBase, junto con la instanciación de una prueba de concepto tomada del área de M&S.

**Abstract.** Modeling dynamic behaviors using events as triggers of state changes is a topic of interest in the Modeling and Simulation (M&S) field. This paper presents the terms, properties, relationships, and axioms of the ParticularEventCO (PEventCO) ontology. Such an ontology is introduced as a semantic model based on events caused by the behavior of entities. PEventCO is located at the core level of the FCD-OntoArch architecture. The foundational level of FCD-OntoArch is supported by the ThingFO ontology. Hence, the main goal is to define a model that allows representing the behavior of dynamic entities following an event-based approach as an additional feature of other ontologies included in FCD-OntoArch. Here, we detail how the main elements of ThingFO are redefined in PEventCO. We also include the formal specification of PEventCO using the ConceptBase software tool and the instantiation of a proof of concept from the M&S field.

**Keywords:** Objetos y eventos, sistemas dinámicos de eventos discretos, fenómeno, modelo semántico, ontología.

Received April 2022; Accepted Jun 2022; Published July 2022

## 1 Introducción

La definición de comportamientos dinámicos usando eventos es un área de interés en Modelado y Simulación (M&S). El M&S basado en eventos discretos parte de la noción de evento, el cual se define como un cambio en el estado del modelo [1]. Comúnmente, los modelos de simulación parten de la abstracción de un fenómeno contextualizado en un mundo particular dado. Luego, la definición de modelos que describan tales fenómenos se convierte en un problema de interés a nivel conceptual que contribuye a la posterior abstracción, formalización e implementación de la simulación asociada [2].

En la literatura existen diversos modelos y ontologías que incorporan el concepto de "evento". Por ejemplo, en [3] se propone un modelo que conceptualiza eventos como entidades. En este trabajo, los autores argumentan que, al menos en los lenguajes orientados a objetos, el modelado de eventos como entidades proporciona beneficios sustanciales. Entre estos beneficios se destacan que, al ser modelados de forma similar a entidades ordinarias, los eventos son instancias de tipos de eventos (una clase especial de tipo de entidad), pueden participar en relaciones, y pueden ser especializados o generalizados, entre otros. Sin embargo, a nivel ontológico, la decisión de modelar eventos como entidades no parece tener fundamentos válidos ya que los eventos no suelen corresponderse con entidades u objetos. Desde una perspectiva diferente, los autores de [4] proponen un modelo donde los eventos se definen conceptualmente en una ontología fundacional denominada UFO-B por su sigla en inglés *Unified Foundational Ontology*. Una ontología fundacional es una ontología que es independiente de cualquier dominio. Luego, UFO-B permite trabajar con eventos al mismo nivel que el resto de los elementos fundacionales definidos en UFO. Sin embargo, en un contexto fundacional diferente, los eventos pueden ser vistos como afirmaciones (en inglés, assertions) que tienen lugar a partir de la existencia de determinadas cosas (las cuales pueden ser particulares o universales).

En este trabajo se presenta una ontología *Core* denominada *ParticularEventCO* (*PEventCO*) que, sobre la base de la ontología fundacional *Thing Foundational Ontology* (*ThingFO*) [5], incorpora la noción de eventos como "assertions" que tienen lugar a causa del comportamiento de las entidades. Ambas ontologías quedan definidas sobre la base de una arquitectura ontológica de cuatro capas llamada FCD-OntoArch (*Foundational*, *Core*, and *Domain Ontological Architecture for Sciences*). Luego, los elementos de *PEventCO* se basan en los términos y relaciones de *ThingFO* con el objetivo de incorporar una representación de comportamiento basada en eventos.

Es importante destacar que este trabajo surge como una extensión del trabajo propuesto en [6], donde se ha presentado la definición de la ontología *PEventCO* junto con los principales lineamientos de FCD-OntoArch y una breve descripción de una prueba de concepto asociada al área de M&S. En esta extensión se incorporan las siguientes modificaciones: *i)* se introduce una actualización de los niveles de la arquitectura FCD-OntoArch en virtud de mejorar su definición, *ii)* se incorporan las preguntas de competencia que han guiado el desarrollo de *PEventCO*, *iii)* se completa la descripción de la implementación de *PEventCO* con la especificación de las preguntas de competencia, y *iv)* se mejora la descripción de la prueba de concepto incorporando mayor detalle en las entidades instanciadas y un Anexo que describe el caso de estudio original.

El resto del trabajo se encuentra estructurado de la siguiente manera. La Sección 2 detalla los fundamentos de la propuesta, describiendo brevemente la perspectiva adoptada. La Sección 3 introduce la actualización de la arquitectura FCD-OntoArch junto con los principales elementos de *ThingFO* que son utilizados como base de la definición de *PEventCO*. La Sección 4 presenta la ontología *PEventCO* detallando las preguntas de competencia que han dado origen a los elementos definidos junto con los términos, relaciones, propiedades y axiomas que la conforman. La Sección 5 describe la implementación del modelo semántico y sus preguntas de competencia en lenguaje Telos haciendo uso de la herramienta ConceptBase. Además, mejora la instanciación previa de la prueba de concepto junto con una discusión de los resultados alcanzados. Finalmente, la Sección 6 se encuentra dedicada a las conclusiones y trabajos futuros.

## 2 Los Objetos y su Relación con los Eventos

En las ciencias aplicadas, los modelos se construyen sobre la base de la abstracción de un fenómeno. Esto es, un modelo es concebido como una representación física, matemática y/o lógica de un sistema, entidad, o proceso sobre el cual se genera una abstracción. Dicha abstracción es posteriormente formalizada e implementada en un lenguaje particular para los fines propios de la ciencia. En este contexto, en el área de M&S, el sistema del mundo real a partir del cual se define un modelo de simulación es considerado el *fenómeno y/o sistema de interés* [7]. De acuerdo con [8], los modelos de simulación suelen asociarse de forma natural al mundo real, mucho más que los sistemas de software, ya que lidian con la representación y manipulación de un conjunto de entidades que forma parte del mundo real (y no con artefactos de implementación asociados). Luego, estas entidades a representar existen en un mundo particular, donde el modelador entiende la realidad según sus propias construcciones mentales.

El M&S basado en eventos discretos se basa en la noción de evento, el cual se define como un cambio en el estado del modelo [1]. Un *evento* se produce en un instante determinado (denominado tiempo del evento) y hace que una parte componente del modelo de simulación se active para producir un cambio de estado (por ejemplo, que se genere un cambio en el valor de un atributo de la entidad representada). Entonces, el estado de una entidad queda definido como el conjunto de valores de todos los atributos que la componen en un instante determinado de tiempo. A través del tiempo, una entidad tendrá múltiples estados como resultado de su evolución temporal. Los *sistemas de interés* al cual pertenecen este tipo de entidades (las cuales pueden representarse con variables discretas sobre una base de tiempo continuo), se denominan Sistemas Dinámicos de Eventos Discretos (DEDS, por su sigla en inglés *Discrete-Event Dynamic Systems*). Por su parte, los diferentes métodos de M&S de DEDS se denominan métodos de M&S de eventos discretos. Aunque en la actualidad existen diversos métodos que dan soporte al M&S de este tipo de sistemas desde el punto de vista de la construcción del modelo de simulación, de acuerdo con [1-2], es necesario definir nuevos métodos que permitan modelar los fenómenos en los cuales las entidades cambian debido a la ocurrencia de eventos particulares y la evolución del *sistema de interés* depende de las interacciones de dichos eventos. Sobre la base de aplicación de tales métodos, es

que los modelos de simulación pueden ser definidos como artefactos asociados que se encuentran expresados en un lenguaje de simulación específico.

Una semántica claramente definida del modelo conceptual de un dominio conduce a una mayor calidad general del modelo construido sobre dicho modelo de dominio, dando lugar a una mejora en el nivel de comprensión, la capacidad de mantenimiento, la interoperabilidad y la capacidad de evolución [8]. El principal beneficio que se obtiene al establecer los fundamentos ontológicos de un dominio es un claro entendimiento de su semántica en el mundo real. Una ontología es una especificación explícita de una conceptualización, es decir, una visión abstracta y simplificada del mundo que incluye los objetos y sus relaciones en un dominio de interés [9]. En este contexto, la utilización de ontologías como mecanismo de soporte para la definición de los DEDS surge naturalmente como una solución aplicable.

El término Simulación de Eventos Discretos (DES, por su sigla en inglés *Discrete Event Simulation*) se ha establecido como un término general que engloba distintos enfoques de simulación. Estos enfoques se basan en la idea general de representar un DEDS modelando su estado como una estructura compuesta de variables de estado y, luego, modelar su dinámica en base a los eventos responsables de los cambios de estado asociados [10]. Pegden [11] caracteriza estos enfoques según una perspectiva denominada "worldviews". El paradigma denominado "event worldview" ve al *sistema de interés* como una serie de eventos instantáneos que cambian su estado en el tiempo. Sin embargo, de acuerdo con [12], este paradigma es ontológicamente incompleto. Esto se debe a que el mundo real consiste esencialmente en objetos y eventos, lo que da lugar a un nuevo paradigma denominado "object-event worldview". Este nuevo paradigma se fundamenta en el hecho de que no es posible concebir los eventos como una serie de ocurrencias aisladas que no tienen ninguna vinculación con los objetos, ya que son los objetos los actores primarios de los eventos. En este contexto, las propiedades de estado de un objeto por medio de sus capacidades son las que influyen en la evolución de su comportamiento y, viceversa, los comportamientos en reacción a acciones externas dan lugar a cambios en las propiedades de los objetos.

Con el objetivo de describir esta dinámica a nivel semántico, en este trabajo se presenta una ontología denominada *PEventCO*. Esta ontología es diseñada como parte de la FCD-OntoArch, donde la ontología fundacional es *ThingFO* [5,13]. De esta manera, los elementos definidos en *PEventCO* incorporan un nuevo componente ontológico de nivel *Core* formulado en base al conjunto de elementos fundacionales, el cual brinda soporte a la existencia de eventos asociados al comportamiento de las entidades.

## 3 Descripción General de la Arquitectura FCD-OntoArch y de la Ontología Fundacional ThingFO

Una arquitectura ontológica debe fomentar la modularidad, extensibilidad y reutilización de los elementos ontológicos en diferentes niveles. Por lo tanto, ninguna ontología debe concebirse y desarrollarse de forma aislada, sino que debe vincularse a una especificación explícita de una arquitectura ontológica en capas. Para *PEventCO*, esta arquitectura ontológica se denomina FCD-OntoArch. En la subsección 3.1, se describen

algunos aspectos de tal arquitectura tales como niveles, guías y reglas a seguir para ubicar y armonizar las ontologías componentes. Luego, en la subsección 3.2, describimos brevemente la ontología fundacional *ThingFO*, detallando principalmente aquellos elementos (términos y relaciones) que son reutilizados y especializados en *PEventCO*.

### 3.1 FCD-OntoArch: Niveles Ontológicos, Guías y Reglas

La arquitectura FCD-OntoArch ha sido propuesta en [5] como una estructura ontológica de cinco niveles conceptuales que da lugar a la definición de distintos tipos de ontologías según el nivel de abstracción al cual pertenecen. Se trata de una estructura basada en capas, en la cual cada capa se corresponde con un nivel de abstracción específico, a saber: *Foundational*, *Core*, *Domain* e *Instance*. El nivel *Domain* se divide en dos subniveles específicos *Top-domain* y *Low-domain*, a fin de permitir distinguir modelos ontológicos de ámbito general y modelos ontológicos de dominios específicos.

Esta arquitectura multicapa promueve el uso de diferentes niveles de abstracción que se deben tener en cuenta al concebir y desarrollar ontologías. Por medio del uso de estos niveles, se da lugar a la correcta ubicación de los modelos conceptuales dentro del esquema general. Por lo tanto, tal como se ha mencionado con anterioridad, la arquitectura fomenta la modularidad, extensibilidad y reutilización de los elementos ontológicos definidos en todos sus niveles.

Sobre la base de los niveles definidos, existen dos lineamientos a seguir:

- *Guía #1*: Cualquier conceptualización/formalización de una ontología desarrollada como un artefacto no puede concebirse aisladamente de una especificación explícita de una arquitectura ontológica en capas. Por ejemplo, una ontología fundacional debe encontrarse en el nivel superior (Nivel Ontológico *Foundational*) de la arquitectura ontológica.
- *Guía #2*: En el Nivel Ontológico *Foundational* de la arquitectura ontológica, para cumplir con los principios de completitud, minimalismo y delegación de preocupaciones, solo se debe tener una ontología fundacional.

En FCD-OntoArch, a fin de brindar una identificación a los componentes o paquetes, las ontologías que pertenecen a un nivel específico tienen una denominación asociada a su ubicación dentro de la arquitectura. Por ejemplo, las ontologías de nivel *Foundational* se denominan *Foundational Ontologies* (FO), mientras que los modelos semánticos de nivel *Core* se denominan *Core Ontologies* (CO), y así sucesivamente.

La Figura 1 presenta la arquitectura FCD-OntoArch. La estructura de paquetes situada a la izquierda de la figura presenta los cinco niveles de abstracción propuestos. La estructura de paquetes en gris y naranja (situada a la derecha de la figura) representa las ontologías incluidas dentro de la arquitectura. A nivel *Foundational*, y siguiendo la Guía #2, los autores proponen una única ontología llamada *ThingFO* [13]. Esto difiere de otras propuestas como UFO [14], la cual se compone de tres ontologías en este nivel (UFO-A, UFO-B, y UFO-C). En cuanto al resto de los niveles, se tiene que las ontologías incluidas son: a nivel *Core*: *ProcessCO* [15], *GoalCO*, *SituationCO*, y *ProjectCO*; a nivel *Top-domain*: *TestTDO* [16], *FRsTDO* (*Functional Requirements*), *NFRsTDO* (*Non-Functional Requirements*), y *MEvalTDO* (*Measurement and Evaluation*); y a nivel *Low-domain*: *MetricsLDO* e *IndicatorsLDO*.

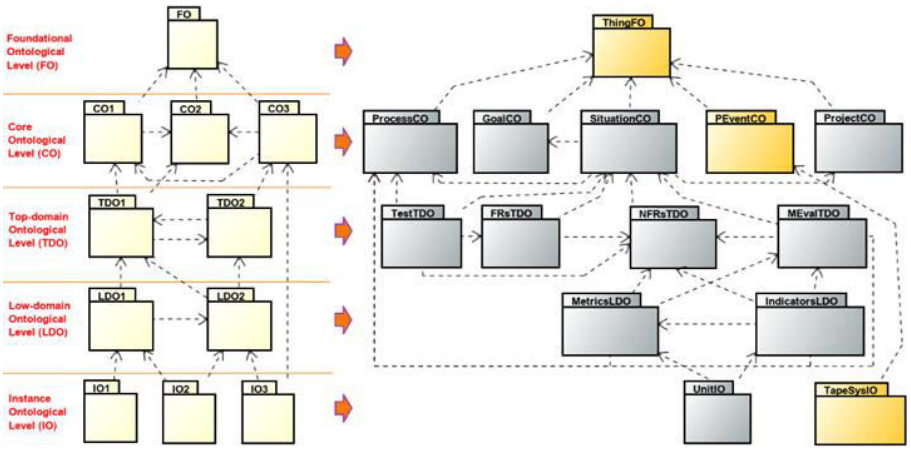


**Fig. 1.** Arquitectura ontológica de cinco niveles que considera los niveles *Foundational*, *Core*, *Top-domain*, *Low-domain* e *Instance*. Los componentes conceptuales desarrollados se muestran en el nivel correspondiente. Tener en cuenta que se utilizan las abreviaturas NFR para *Non-Functional Requirements*, FR para *Functional Requirements*, MEval para *Measurement and Evaluation*, PEvent para *Particular Event* y TapeSys para *Tape System.*

Al usar los niveles ontológicos anteriores, la ubicación correcta de las ontologías dentro del esquema general debe establecerse mediante reglas, a saber:

- *Regla #1*: Toda ontología situada en el nivel CO, TDO, o LDO de la arquitectura ontológica representada en la Figura 1 debe garantizar una correspondencia de sus elementos con los elementos definidos en el nivel inmediatamente superior. Por ejemplo, para introducir una nueva ontología a nivel *Core* se debe garantizar que sus elementos tengan una correspondencia con los elementos definidos en el nivel *Foundational*. Esto permite que los términos y relaciones de las ontologías de los niveles inferiores se enriquezcan semánticamente con los términos y relaciones de las ontologías de niveles superiores.
- *Regla #2*: Las ontologías de un mismo nivel (excepto en los niveles FO e IO) pueden relacionarse entre sí, pero debe garantizarse que en su definición conjunta (como un todo) no se violen los principios del nivel inmediato superior. Esto implica que, si una ontología de nivel *Core* utiliza elementos de otra ontología del mismo nivel, en conjunto ambos modelos deben garantizar una correcta definición respecto a la ontología fundacional. Esto permite que los términos y relaciones de las ontologías de un mismo nivel puedan complementarse entre sí, manteniendo una correspondencia con las definiciones de las ontologías de los niveles superiores.
- *Regla #3*: En el nivel ontológico *Instance*, sólo se pueden ubicar individuos de cosas particulares. Una cosa representada en el nivel fundacional como una clase particular o cualquiera de sus subclases (con la semántica de *Thing*) representada apropiadamente en los niveles inferiores da como resultado instancias. Por lo tanto, un individuo es una instancia de una clase particular en niveles superiores.

En este contexto, la ontología *PEventCO* (paquete resaltado en naranja, en la parte derecha de la Figura 1) se encuentra situada a nivel *Core*. En la Sección 4 se demuestra la forma en la cual *PEventCO* se construyó teniendo en cuenta la Regla #1. Además, en la Sección 5 se presentan instancias de *PEventCO* utilizando un caso clásico de máquina de Turing. Tales instancias se definen en *TapeSysIO* (paquete naranja en la estructura derecha de la Figura 1) en el Nivel Ontológico *Instance* a fin de cumplir con la Regla #3. Finalmente, cabe mencionar que en este artículo no discutiremos cómo *PEventCO* contribuye al logro de la Regla #2. Esto se debe a que es necesario investigar más a fondo su interacción con otras ontologías del mismo nivel, como por ejemplo la interacción de *PEventCO* con *SituationCO*. Por lo tanto, este aspecto forma parte de un futuro trabajo de investigación.

### 3.2 Descripción de algunos Elementos de la Ontología Fundacional ThingFO

Las ontologías fundacionales son representaciones de conceptos de alto nivel, que son independientes de cualquier dominio. En este contexto, *ThingFO* ha sido propuesta como una ontología de nivel *Foundational* en FCD-OntoArch. Esta ontología cuenta con un conjunto reducido de términos y relaciones que refieren tanto a elementos particulares como a universales del mundo. Su objetivo es identificar el conjunto mínimo de términos, propiedades, relaciones y restricciones que representen el mundo, permitiendo que los mismos sean reutilizados y/o especializados en los dominios de las distintas ciencias. Por lo tanto, en su concepción se tuvo en cuenta a la Guía #2.

Una primera versión de *ThingFO* ha sido presentada en [5] y luego actualizada en [13]. Como resultado, su versión actual (v1.3) cuenta con 19 términos definidos, 10 propiedades definidas, 3 axiomas especificados en lógica de primer orden y 12 relaciones no taxonómicas definidas que están bien equilibradas con las relaciones taxonómicas. Se puede acceder a sus definiciones y representación gráfica en UML en http://arxiv.org/abs/2107.09129.

Los principales términos incluidos en esta ontología son: *Thing*, *Property* y *Power* para particulares, *Thing Category* para universales y *Assertions* como elemento clave para representar las afirmaciones que surgen cuando un agente humano representa y modela intencionadamente elementos del mundo que corresponden tanto a cosas particulares (*Assertion on Particulars*) como a universales (*Assertion on Universals*).

La Tabla 1 contiene las definiciones y algunas notas para los 7 términos de *ThingFO* que enriquecen semánticamente los términos de *PEventCO*. Además, la Tabla 2 contiene las definiciones de las 6 relaciones no taxonómicas de *ThingFO* que se especializan en *PEventCO*.

## 4 PEventCO: Ontología Core en FCD-OntoArch

En virtud de presentar la ontología *PEventCO*, en la subsección 4.1 se detallan las preguntas de competencia utilizadas para su construcción. En la subsección 4.2, se introducen los términos, propiedades y relaciones definidos en la ontología. Finalmente, la subsección 4.3 detalla los axiomas que complementan los elementos de *PEventCO*.

**Tabla 1** Definiciones de los términos incluidos en *ThingFO* que se reutilizan en *PEventCO*.

| Término | Definición |
|---|---|
| **Thing** (traducción: **Cosa**) | Clase o tipo de objeto perceptible o concebible, o sus individuos, de un mundo particular dado.<br>Nota 1: Una Cosa como clase representa e implica individuos o instancias únicas, no una categoría universal. Por lo tanto, una Cosa particular resulta en instancias, mientras que una Cosa universal (es decir, una Categoría de Cosa) no resulta en instancias, al menos con un significado valioso de individuo en un mundo particular dado.<br>Nota 2: Una subclase de una Cosa particular, como una clase particular, puede ser representada en cualquier nivel inferior de la arquitectura ontológica (representada en la Figura 1), excepto en el Nivel Ontológico *Instance*, donde solo hay o existen individuos particulares (Regla #3). |
| **Property** (traducción: **Propiedad**) | Se refiere a la constitución intrínseca, estructura o partes de un Cosa particular.<br>Nota: Una Propiedad es un miembro de la tríada que conforma la identidad única denominada Cosa. |
| **Power** (traducción: **Capacidad**) | Se refiere a lo que una Cosa particular hace, puede hacer o comportase.<br>Note 1: La Capacidad es un miembro de la tríada que conforma la identidad única denominada Cosa.<br>Note 2: Según [17]: "*Powers are the way of acting of a things' properties; powers are a things' properties in action*" y "*Things have properties, these properties instantiate […] acting powers, and this ensemble of things, properties and powers cause any events that might occur*". |
| **Assertion** (traducción: **Aserción**) | Clase o tipo que representa una declaración o expresión positiva y explícita que alguien hace sobre algo concerniente a las Cosas, o sus categorías, con base en pensamientos, percepciones, hechos, intuiciones, intenciones y/o creencias, concebida con el intento de proporcionar evidencia actual o ulterior.<br>Nota 1: La parte de la definición anterior que indica "…declaración o expresión positiva y explícita que alguien hace…" significa que un ser humano concreto -como Cosa particular- define o concibe Aserciones.<br>Nota 2: Con respecto a una Cosa particular o categoría, una declaración positiva se refiere a lo que es, fue o será, y no contiene ninguna indicación de aprobación o desaprobación.<br>Nota 3: ISO 21838-1 define "expresión" como "*word or group of words or corresponding symbols that can be used in making an assertion*"[18]. |
| **Assertion on Particulars** (traducción: **Aserción sobre Particulares**) | Es una Aserción que alguien hace sobre algo de una o más Cosas particulares. |
| **Action-related Assertion** (traducción: **Aserción relacionada a Acción**) | Es una Aserción relacionada a la interacción y el acontecer de las Cosas ya que las Capacidades actuantes provocan cualquier evento que pueda ocurrir.<br>Nota: Las Cosas particulares pueden interactuar entre sí, al igual que una Cosa puede actuar sobre sí misma. |
| **Time-related Assertion** (traducción: **Aserción relacionada al Tiempo**) | Es una Aserción relacionada con el tiempo visto como una Cosa, o sus Propiedades o Capacidades, lo que puede implicar especificar fronteras o límites temporales, entre otros aspectos, para distintas situaciones y eventos. |

**Tabla 2** Definiciones de aquellas relaciones no taxonómicas incluidas en *ThingFO* que están especializadas en *PEventCO*.

| Relación | Definición |
|---|---|
| **acts upon** | Una Capacidad actúa sobre una o más Propiedades, por lo que puede conocer o actualizar el estado de las propiedades de la Cosa.<br>Nota: Esta relación representa acciones internas, es decir, sobre la misma Cosa, no sobre otras Cosas. Esta restricción se especifica mediante un axioma en http://arxiv.org/abs/2107.09129. |
| **deals with particulars** | Una Aserción sobre Particulares trata con Cosas particulares, tanto clases y subtipos como instancias. |
| **defines** | Una Cosa define ninguna o muchas Aserciones.<br>Nota: Por ejemplo, una Cosa particular como un agente humano define o concibe Aserciones, tales como metas, eventos y situaciones, entre muchas otras. |
| **enables** | Una Propiedad habilita las Capacidades de una Cosa particular.<br>Nota: Debido a que las Propiedades de una Cosa están presentes, el comportamiento de la entidad puede habilitarse y manifestarse. Esta relación está restringida por un axioma en http://arxiv.org/abs/2107.09129. |
| **interacts with other** | Debido a las Capacidades de una Cosa, las Cosas particulares interactúan entre sí.<br>Nota: Esta relación representa acciones sobre otras Cosas, no sobre la misma Cosa. Esta restricción está especificada por un axioma en http://arxiv.org/abs/2107.09129. |
| **relates with** | Una Cosa se relaciona con otra Cosa en particular. |

### 4.1 Requisitos de Alcance: Preguntas de Competencia

Con el objetivo de determinar el alcance de la ontología *PEventCO*, se definieron preguntas de competencia. Una pregunta de competencia es una pregunta que la ontología bajo desarrollo debe ser capaz de responder. Estas preguntas sirven como prueba de control de calidad, ya que permiten determinar si la ontología resultante contiene suficiente información como para responder el conjunto de preguntas planteadas de la forma esperada.

La Tabla 3 resume el conjunto de preguntas de competencia formuladas para la ontología *PEventCO*. Como puede observarse, estas preguntas han sido clasificadas según el tipo de respuesta esperada. Las categorías utilizadas en esta clasificación corresponden a *evento*, *propiedad de estado*, *entidad*, y *comportamiento*. En este sentido, las preguntas CQ1 a CQ4 refieren a cuáles son los *eventos* asociados a algún otro tipo de elemento del dominio. Por ejemplo, la pregunta CQ1 refiere a los *eventos* vinculados al *estado de una entidad*. Por su parte, las preguntas CQ5 y CQ6 corresponden a *propiedades de estado* asociadas, respectivamente, a eventos y entidades. Las preguntas CQ7 a CQ10 refieren a la existencia de entidades vinculadas a distintos tipos de eventos (como ser, eventos de externos, de entrada y de salida). Finalmente, la pregunta CQ11 busca determinar comportamientos asociados a propiedades de estado.

**Tabla 3.** Listado de las preguntas de competencia utilizadas para definir el alcance de la ontología de nivel CO denominada *PEventCO*. La versión en inglés se detalla en el Apéndice A.

| Categoría | Id. | Descripción |
|---|---|---|
| **Evento** | **CQ1** | ¿Cuáles son los *eventos* internos que actúan sobre el estado de la entidad? |
| | **CQ2** | ¿Cuáles son los *eventos* externos por los que reacciona la entidad? |
| | **CQ3** | ¿Cuáles son los *eventos* externos que produce la entidad? |
| | **CQ4** | ¿Cuáles son los *eventos* causados por la entidad? |
| **Propiedad de Estado** | **CQ5** | ¿Cuáles son las *propiedades de estado* sobre las que actúa el evento? |
| | **CQ6** | ¿Cuáles son las *propiedades de estado* de la entidad? |
| **Entidad** | **CQ7** | ¿Cuáles son las *entidades* influenciadas por eventos externos de la entidad? |
| | **CQ8** | ¿Cómo la *entidad* es influenciada por eventos externos? |
| | **CQ9** | ¿Es la *entidad* un consumidor de sus eventos de entrada? |
| | **CQ10** | ¿Es la *entidad* un productor de eventos de salida? |
| **Comportamiento** | **CQ11** | ¿Cuáles son los *comportamientos* que actúan sobre la propiedad de estado? |

### 4.2 Términos, Propiedades y Relaciones

En la Figura 2 se presenta el conjunto de términos, propiedades y relaciones que compone la ontología *PEventCO*. Tales elementos se describen haciendo uso de un diagrama de clases UML. Es importante notar que, dada la arquitectura *FCD-OntoArch*, debe existir una correspondencia entre los elementos que conforman el modelo de *PEventCO* y los elementos definidos en *ThingFO* (Regla #1). Esto es válido tanto para los términos como así también para las relaciones y propiedades. A continuación, se describe la forma en la cual se vinculan elementos de distintos niveles.

De acuerdo con los lineamientos planteados en [15], cuando UML es utilizado como lenguaje para generar modelos ontológicos, es posible utilizar estereotipos como mecanismo de enriquecimiento semántico de los nuevos conceptos. Esto es, un estereotipo *<<stereotype>>* aplicado sobre una nueva clase *Class* actúa como vínculo entre el nuevo término modelado por *Class* y el término existente modelado por *<<stereotype>>* a fin de indicar que este último enriquece semánticamente al nuevo elemento. El diagrama de clases UML propuesto en la Figura 2 hace uso de este mecanismo con el objetivo de establecer un vínculo entre los términos definidos a nivel CO (esto es, las nuevas clases UML definidas en *PEventCO*) y los términos situados a nivel FO (es decir, las clases existentes en *ThingFO*). Es importante notar que un nuevo término puede ser enriquecido semánticamente por más de un término existente.

En *PEventCO*, una *Entity* representa un objeto tangible o intangible para el cual se define un comportamiento dinámico. Luego, *Entity* tiene semántica de *Thing* (del nivel FO) ya que dicho comportamiento es descripto haciendo uso de sus *Powers* and *Properties*. Específicamente, las *Powers* que componen una *Entity* se definen como *Behaviors*. Esto permite considerar las capacidades de las entidades como comportamiento

asociado a las mismas. A su vez, las *Properties* que componen una *Entity* se definen como *State Properties*. De esta manera, las entidades quedan definidas según las propiedades asociadas a su estado. En conjunto, ambas definiciones dan lugar a centrar la atención en la forma en la cual los *Behaviors* actúan en las *State Properties*. De acuerdo con Fleetwood [17], las capacidades (*Powers*) son la forma de actuar de las propiedades de las cosas (es decir, las capacidades son las propiedades de las cosas en acción). Luego, este conjunto de cosas, propiedades y capacidades causa cualquier evento que pueda ocurrir. En *PEventCO*, los eventos son modelados a partir del concepto *Particular Event*. Así, la tupla *Entity-Behavior-State Property-Particular Event* es la base para conceptualizar la existencia de eventos.

Todos los eventos se corresponden con un *Particular Event*. En este contexto, un evento particular tiene semántica de *Assertion on Particulars* y *Action-related Assertion* de *ThingFO*. De esta manera, un evento particular declara y especifica (como una aserción) la ocurrencia de una acción en una entidad. Se relaciona con la forma en la cual las entidades interactúan, ya que los comportamientos causan cualquier evento que tenga lugar. Es decir, un evento particular establece la ocurrencia de la acción de algún objeto concreto en el contexto de un lugar determinado durante un intervalo o instante de tiempo particular. En este sentido, los mecanismos de eventos pueden requerir considerar límites temporales. Luego, un *Time Boundary* tiene semántica de *Assertion on Particulars* y *Time-related Assertion* de *ThingFO* con el objetivo de especificar las restricciones temporales a partir de las cuales se dan eventos particulares.

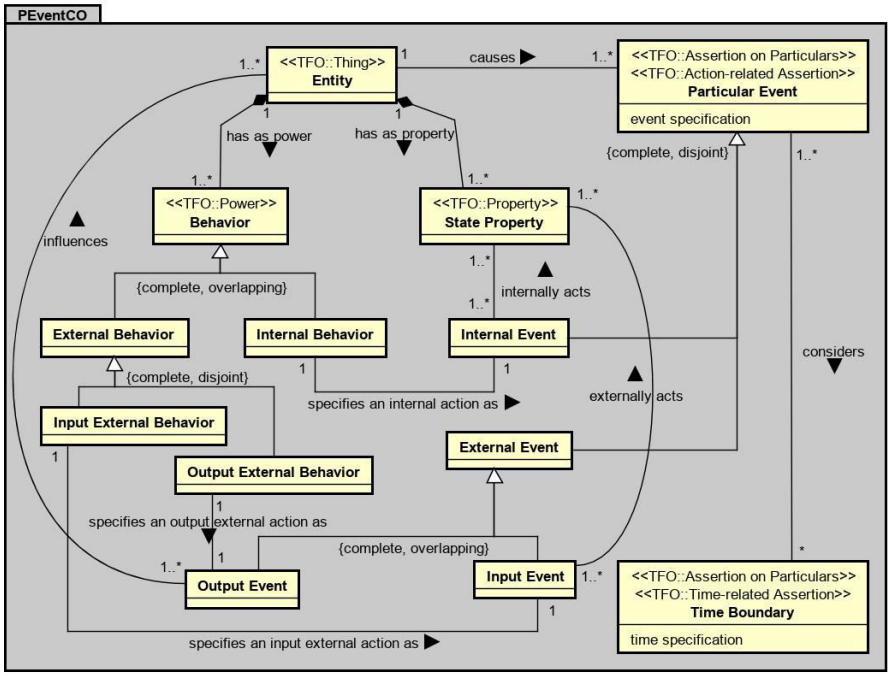


**Fig. 2.** Diagrama UML de los términos, relaciones y propiedades que componen la ontología *PEventCO*. Los estereotipos refieren a términos definidos en *ThingFO* (TFO).

En *PEventCO*, existen dos tipos de *Behaviors* que pueden formar parte de una *Entity*, estos son: *Internal Behavior* y *External Behavior*. Un *Internal Behavior* representa un comportamiento que refiere a lo que una entidad particular puede hacer para actuar sobre su estado. Los comportamientos internos actúan sobre las *State Properties* por medio de acciones internas que dan lugar a *Internal Events*. Es decir, un comportamiento puede modificar y/o consultar las propiedades de estado haciendo uso de eventos internos que se desprenden de su definición. Por su parte, un *External Behavior* representa un comportamiento que refiere a lo que una entidad particular puede hacer para influenciar otras entidades (*Output External Behavior*) y/o en respuesta a la influencia de otras entidades (*Input External Behavior*). Los comportamientos externos dan lugar a acciones externas que se presentan como *External Events*. Existen dos tipos de *External Events*, a saber: *Input Event* y *Output Event*. Un *Input Event* es un evento externo asociado a un comportamiento externo de entrada que refiere a la ocurrencia de una acción externa que actúa sobre las propiedades de estado de una entidad. Esto es, un evento de entrada se asocia a la forma en la cual la entidad reacciona ante estímulos externos por medio de un comportamiento propio definido. Este evento, actúa sobre las propiedades de estado ya sea modificando o consultando su estructura. En contraposición, un *Output Event* es un evento externo que declara y especifica explícitamente la ocurrencia de una acción externa de salida en una entidad (a partir de un comportamiento externo) que busca generar algún impacto sobre las propiedades de otras entidades. En este punto, es importante comprender que una entidad no puede afectar a otra de forma directa. Son las capacidades (en este caso, los comportamientos) de una entidad los que actúan sobre sus propiedades de estado. Luego, cuando una entidad afecta a otra lo hace por medio de un evento de salida. Este evento es producido por el comportamiento externo de la entidad origen de la interacción. En respuesta, la entidad destino actúa en consecuencia según su propio comportamiento externo asociado al evento de entrada. Luego, un evento de entrada puede ser visto como el estímulo que necesita una entidad para actuar en consecuencia a la influencia de otra entidad externa.

Las Tablas 4 a 6 presentan, respectivamente, la definición de cada término, propiedad y relación que forma parte de *PEventCO*. El Apéndice B contiene la versión en inglés de estas definiciones.

### 4.3 Axiomas

En *PEventCO* se definen 13 axiomas que ayudan a dar consistencia a las definiciones propuestas. La Tabla 7 resume el conjunto de axiomas asociados al modelo propuesto en la Figura 2 haciendo uso de lógica de primer orden. Estos axiomas se clasifican en dos grupos, a saber: *i) Axiomas entre niveles*: Son los axiomas que garantizan la correspondencia entre las nuevas relaciones de nivel CO y las relaciones existentes en el nivel FO. Este grupo incluye los axiomas A1 a A8; *ii) Axiomas de nivel CO*: Además de dar consistencia entre niveles, estos axiomas dan consistencia entre los elementos definidos dentro de *PEventCO*. Este grupo incluye los axiomas A9 a A13.

El axioma A1 define que una *Entity* que causa un *Particular Event* es una *Thing* que define una *Assertion on Particulars*. Luego, la relación *causes* del nivel CO redefine la

relación *defines* de nivel FO, donde *Entity* actúa como *Thing* y *Particular Event* como *Assertion on Particulars*. Lo mismo ocurre con *hasAsPower* y *hasAsProperty* respecto a las composiciones *Thing-Power* y *Thing-Property*. Ambos vínculos se esquematizan en los axiomas A2 y A3, respectivamente. El axioma A4, redefine la relación *relatesWith* de nivel FO entre dos *Assertions* como *considers* para el caso de las aserciones *ParticularEvent* y *TimeBoundary*.

En este mismo contexto, los axiomas A5 a A7 definen la forma en la cual las relaciones *actsUpon* e *interactsWithOther* del nivel FO pueden ser inferidas a partir de lo modelado a nivel CO. De acuerdo con el nivel FO, la relación *actsUpon* entre *Power* y *Property* es definida como la vinculación que se establece cuando "un *Power* actúa sobre una o más *Properties*, pudiendo consultarlas o actualizar su estado". Luego, esta relación ayuda a representar los eventos que son producidos por un comportamiento cuyo efecto actuará sobre las propiedades de estado. En este sentido, hay dos tipos de eventos que actúan sobre las propiedades de estado: eventos internos y eventos de entrada. El axioma A5 define la forma en la cual las relaciones denominadas *specifiesAnInternalActionAs* e *internallyActs* entre *InternalBehavior*, *InternalEvent* y *StateProperty*, componen la relación *actsUpon* de nivel superior. Lo mismo ocurre en el axioma A6 con las relaciones *specifiesAnInputExternalActionAs* y *externallyActs* entre *InputExternalBehavior*, *InputEvent* y *StateProperty*. Por su parte, en el nivel FO, la relación *interactsWithOther* se define como el vínculo que se establece cuando, debido al *Power* de una *Thing*, las cosas interactúan entre sí. Esta interacción ayuda a representar eventos de salida que son producidos por el comportamiento de una entidad pero que, a su vez, buscan afectar a otras entidades. El axioma A7 define la forma en la cual las relaciones *specifiesAnOutputExternalActionAs* e *influences* entre *OutputExternalBehavior*, *OutputEvent* y *Entity* dan lugar a *interactsWithOther*.

**Tabla 4.** Definición de los términos de *PEventCO* que han sido especificados en la Figura 2 haciendo uso de clases UML. Se detallan los sinónimos a fin de dar consistencia a la definición.

| Término | Definición |
|---|---|
| **Entity** (sinónimo: **Dynamic Entity)** | Objeto particular o concreto, tangible o intangible, para el que se define de forma explícita un comportamiento dinámico con sus propiedades y capacidades. |
| **Behavior** | Refiere a la forma en la que se comporta una entidad en particular bajo condiciones establecidas. |
| **State Property** | Refiere a la estructura de estado intrínseca de una determinada entidad. |
| **Particular Event** | Aserción que declara y especifica explícitamente la ocurrencia de una acción en una entidad. |
| **Time Boundary** | Es una aserción relacionada con el tiempo que especifica los límites y restricciones temporales a partir de los cuales se dan los eventos particulares. |
| **Internal Behavior** (sinónimo: **Internal Power**) | Es un tipo de comportamiento que establece y especifica la ocurrencia de acciones internas en una entidad. |

| | |
|---|---|
| **External Behavior** (sinónimo: **External Power**) | Es un tipo de comportamiento que establece y especifica la ocurrencia de acciones externas sobre una entidad. |
| **Input External Behavior** (sinónimo: **Input Behavior**) | Es un comportamiento externo que establece y especifica la ocurrencia de acciones de entrada sobre una entidad. |
| **Output External Behavior** (sinónimo: **Output Behavior**) | Es un comportamiento externo que establece y especifica la ocurrencia de acciones de salida sobre una entidad. |
| **Internal Event** (sinónimo: **Entity Internal Event**) | Evento particular que establece y especifica explícitamente la ocurrencia de una acción interna en una entidad que actúa sobre sus propiedades de estado. |
| **External Event** (sinónimo: **Entity External Event**) | Evento particular que declara y especifica explícitamente la ocurrencia de una acción externa en una entidad. |
| **Input Event** (sinónimo: **External Input Event**) | Es un evento externo que declara y especifica explícitamente la ocurrencia de una acción externa que actúa sobre las propiedades de estado de una entidad. |
| **Output Event** (sinónimo: **External Output Event**) | Es un evento externo que declara y especifica explícitamente la ocurrencia de una acción externa en una entidad que tiene algún impacto sobre otras entidades. |

**Tabla 5.** Definición de las propiedades de los términos incluidos en *PEventCO* que han sido especificadas en la Figura 2 haciendo uso de atributos de clases UML. En este caso, estos atributos redefinen la propiedad "specification" de "Assertion".

| Término | Propiedad | Definición |
|---|---|---|
| **Particular Event** | **event specification** | Especifica el evento en un lenguaje dado. |
| **Time Boundary** | **time specification** | Especifica las relaciones y restricciones temporales para los eventos, ya que los eventos ocurren en el tiempo. |

**Tabla 6.** Definición de las relaciones no taxonómicas de *PEventCO* que han sido especificadas en la Figura 2 haciendo uso de asociaciones UML. La correspondencia de estas relaciones con las de nivel FO se presenta en los axiomas de la subsección 4.3.

| Relación | Definición |
|---|---|
| **has as power** | Una entidad tiene uno o más comportamientos que definen sus capacidades. |
| **has as property** | Una entidad tiene una o más propiedades de estado que definen sus propiedades estructurales. |
| **causes** | Una entidad dinámica causa uno o más eventos particulares. |
| **specifies an internal action as** | Un comportamiento interno especifica un evento interno. |
| **specifies an input external action as** | Un comportamiento de entrada externo especifica un evento de entrada externo. |

| | |
|---|---|
| **specifies an output external action as** | Un comportamiento de salida externo especifica un evento de salida externo. |
| **internally acts** | Un evento interno actúa internamente sobre propiedades de estado. |
| **externally acts** | Un evento de entrada actúa externamente sobre propiedades de estado. |
| **considers** | Un evento particular puede requerir de límites temporales. |
| **influences** | Un evento de salida influencia a una o más entidades. |

**Tabla 7.** Definición de los axiomas que complementan los términos y relaciones de la ontología *PEventCO* haciendo uso de lógica de primer orden.

| Id. | Definición |
|---|---|
| **A1** | *∀e,∀ev:[ Entity(e) ^ ParticularEvent(ev) ^ causes(e,ev) → defines(e,ev)]* |
| **A2** | *∀e,∀b: [Entity(e) ^ Behavior(b) ^ hasAsPower(e,b) → partOf(b,e)]* |
| **A3** | *∀e,∀s: [Entity(e) ^ StateProperty(s) ^ hasAsProperty(e,s) → partOf(s,e)]* |
| **A4** | *∀e,∀ib:[ParticularEvent(e)^ TimeBoundary(tb) ^ considers(e,tb) → relatesWith(e,tb)]* |
| **A5** | *∀ib,∀ie,∀sp: [InternalBehavior(ib) ^ specifiesAnInternalActionAs(ib,ie) ^ InternalEvent(ie) ^ internallyActs(ie,sp) ^ StateProperty(sp) → actsUpon(ib,sp)]* |
| **A6** | *∀eb,∀ie,∀sp: [InputExternalBehavior(eb) ^ specifiesAnInputExternalActionAs(eb,ie) ^ InputEvent(ie) ^ externallyActs(ie,sp) ^ StateProperty(sp) → actsUpon(eb,sp)]* |
| **A7** | *∀eb,∀oe,∀e: [OutputExternalBehavior(eb) ^ OutputEvent(oe) ^ Entity(e) ^ specifiesAnOutputExternalActionAs(eb,oe) ^ influences(oe,e) → interactsWithOther(eb,e)]* |
| **A8** | *∀oe,∀e: [OutputEvent(oe) ^ influences(oe,e) ^ Entity(e) → dealsWithParticulars(oe,e)]* |
| **A9** | *∀e,∀ib,∀ie:[Entity(e) ^ partOf(ib,e) ^ InternalBehavior(ib) ^ specifiesAnInternalActionAs(ib,ie)^ InternalEvent(ie) → causes(e,ie) ^ dealsWithParticulars(ie,e)]* |
| **A10** | *∀e,∀eb,∀ie: [Entity(e) ^ partOf(eb,e) ^ InputExternalBehavior(eb) ^ InputEvent(ie) ^ specifiesAnInputExternalActionAs(eb,ie) → dealsWithParticulars(ie,e)]* |
| **A11** | *∀e,∀eb,∀oe: [Entity(e) ^ partOf(eb,e) ^ OutputExternalBehavior(eb) ^ OutputEvent(oe) ^ specifiesAnOutputExternalActionAs(eb,oe) → causes(e,oe) ^ ¬dealsWithParticulars(oe,e)]* |
| **A12** | *∀e,∀tb,∀en: [ParticularEvent(e) ^ TimeBoundary(tb) ^ considers(e,tb) ^ Entity(en) ^ dealsWithParticulars(e,en) → dealsWithParticulars(tb,en)]* |
| **A13** | *∀e,∀en: [OutputEvent(e) ^ influences(en) ^ Entity(en) → ∃eb: [InputExternalBeahvior(eb) ^ hasAsPower(en,eb) ^ specifiesAnInputExternalActionAs(eb, e) ^ InputEvent(e)]]* |

El axioma A8 busca redefinir la relación *dealsWithParticulars* para el caso de *influences* entre *OutputEvent* y *Entity*.

Los axiomas A9 a A11 se encuentran vinculados a la forma en la cual los eventos son causados por una entidad. Específicamente, el axioma A9 indica que todo *InternalEvent* producto del *InternalBehavior* de una *Entity*, es causado por dicha *Entity* y, además, a nivel FO, define la relación *dealsWithParticulars*. El axioma A10 efectiviza la

definición de *dealsWithParticulars*, pero para un *InputEvent* especificado a partir de un *InputExternalBehavior*. Sin embargo, para el caso de *OutputEvent*, aun cuando el evento es causado por la *Entity*, no se lo vincula a nivel FO por medio de la relación *dealsWithParticulars* con dicha *Entity* (axioma A11). Esto se debe a que un *OutputEvent* no puede afectar a la *Entity* a partir de la cual se lo ha emitido.

El axioma A12 tiene como objetivo dar coherencia a la relación *dealsWithParticulars* de nivel FO una vez establecido un vínculo a nivel CO para *ParticularEvent*. Este axioma detalla que el *TimeBoundary* de un *ParticularEvent* que se encuentra vinculado a nivel FO por *dealsWithParticulars* a una *Entity*, también debe estar vinculado por *dealsWithParticulars* a dicha *Entity*. Esto se debe a que el *TimeBoundary* se encuentra vinculado a la *Entity* por medio del *ParticularEvent*, por lo que también debe asociarse en la relación *dealsWithParticulars*.

Finalmente, el axioma A13 permite integrar de manera adecuada los eventos de entrada con los eventos de salida de distintas entidades. Este axioma especifica que una *Entity*, influenciada por un *OutputEntity*, posee un comportamiento externo (*InputExternalBehavior*), el cual está definido por un *InputEvent*.

# 5 Implementación y Prueba de Concepto de PEventCO

A fin de comprobar que el modelo semántico propuesto es satisfacible, en la subsección 5.1 se presenta la especificación de la ontología *PEventCO* en el lenguaje Telos y su implementación en ConceptBase. A partir de la implementación obtenida, en la subsección 5.2, se instancia la ontología propuesta. Finalmente, en la subsección 5.3, se presenta una discusión de los resultados alcanzados.

## 5.1 Especificación en ConceptBase

Con el objetivo de generar instancias del modelo semántico propuesto como *PEventCO* y, teniendo en cuenta que su definición depende de *ThingFO*, ambas ontologías fueron implementadas adoptando el lenguaje Telos. Telos es un lenguaje de modelado conceptual para representar el conocimiento sobre sistemas de información basado en los conceptos centrales de tecnología orientada a objetos, restricciones de integridad y reglas deductivas. Por lo tanto, se empleó el lenguaje Telos para especificar formalmente los conceptos y relaciones introducidos en la subsección 4.2, los axiomas del modelo propuesto en la subsección 4.3, y las preguntas de competencia definidas en la subsección 4.1. Dicha implementación fue realizada utilizando la herramienta ConceptBase [19], la cual es un administrador de base de datos deductivo orientado a objetos que, en nuestro caso, facilita la verificación y validación del modelo propuesto.

En la Figura 3 se ilustra la especificación Telos del concepto *Particular Event* (Tabla 4). En este lenguaje, cada uno de los axiomas modelados para deducir nuevos hechos en la ontología *PEventCO*, han sido definidos como reglas (*rules*). Por ejemplo, la expresión del axioma A4 (Tabla 7) ha sido definida en la regla “rule A4_r” (Figura 3).

Las preguntas de competencia fueron definidas como clases de consultas genéricas (*GenericQueryClass*). Una clase de consulta permite definir un conjunto de objetos de

una clase determinada mediante la especificación de una restricción. La restricción se interpreta como una condición suficiente para la pertenencia a la clase, es decir, todas las instancias que coinciden con la definición de la clase de consulta y cumplen la restricción se consideran instancias de la clase de consulta [19]. Una clase de consulta genérica es una clase de consulta parametrizada. En la Figura 4 se incluye la definición de la pregunta de competencia CQ2 - ¿Cuáles son los eventos externos por los que reacciona la entidad? (Tabla 3). Aquí, *CQ2* se define como una especialización de *External_Event* ya que sus respuestas serán eventos externos y está parametrizada por la entidad (*entity*). En la restricción "cq2_c" se especifica la condición suficiente para que un evento externo sea incluido como respuesta a esta pregunta de competencia. De forma similar, se definieron las preguntas de competencia detalladas en la Tabla 3.

Telos Editor -- localhost:4001

```
1  Particular_Event in Class isA Assertion_on_Particulars,Action_related with
2    attribute
3     event_specification : String;
4     considers : Time_Boundary
5    constraint
6     causes_c : $ forall pe/Particular_Event e1,e2/Entity (e1 causes pe) and (e2 causes pe) ==> (e1 = e2) $;
7     causes_c2 : $ forall pe/Particular_Event exists e/Entity (e causes pe) $;
8     considers_c : $ forall t/Time_Boundary exists pe/Particular_Event ( pe considers t) $
9    rule
10    A4_r : $ forall pe/Particular_Event tb/Time_Boundary (pe considers tb) ==> (pe relates_with tb) $
11 end
12
```

**Fig. 3.** Definición en Telos del término "Particular Event" junto con los axiomas asociados. Estos axiomas han sido definidos como "reglas".

Telos Editor -- localhost:4001

```
1  CQ2 in GenericQueryClass isA External_Event with
2    parameter
3     entity : Entity
4    constraint
5     cq2_c : $ exists sp/State_Property (~this in Input_Event) and (~this externally_acts sp) and
6                                                                    (~entity has_as_property sp) $
7  end
8
```

**Fig. 4.** Definición en Telos de la pregunta de competencia "CQ2: ¿Cuáles son los eventos externos por los que reacciona la entidad?".

### 5.2 Instanciando PEventCO: The Tape System

Para comprobar que el modelo semántico propuesto es satisfacible (junto con la ontología fundacional *ThingFO*), se ha instanciado un caso clásico empleado en el área de M&S [7]: una máquina de Turing construida a partir de dos entidades denominadas TM Control y Tape System. Una breve descripción del caso se incluye en el Apéndice C.

La Figura 5 se centra en la representación del elemento "tape" que refiere a la entidad Tape System. En la parte inferior de la figura se ilustran las instancias. Los arcos sólidos entre instancias representan relaciones explícitamente enunciadas en la base de datos. A su vez, los arcos en líneas punteadas refieren a relaciones inferidas a partir de axiomas (reglas).

El “tape” está compuesto por tres propiedades: cinta del tape (elemento “tape_i_SP”), posición de la cabeza lectora (elemento “pos_SP”), y próximo movimiento que debe realizar la cabeza sobre la cinta (elemento “mov_SP”). El comportamiento del “tape” se define mediante el comportamiento aceptación de un símbolo (definido como “accept_symbol_B”) y el comportamiento asociado al movimiento de la cabeza (definido como “move_head_B”). Ambos comportamientos se especifican a partir de los eventos identificados en el fenómeno modelado.

En particular, se tiene que “accept_symbol_B” es especificado por el evento de entrada “inputSymbol_ExtEv”, el cual permite actuar sobre las propiedades del “tape”: escribe el símbolo en la posición “pos_SP” actual de la cinta “tape_SP” y, además, fija el siguiente movimiento de la cabeza “mov_SP”. Este evento de entrada desencadena los eventos que especifican a “move_head_B”. Estos son: *i)* el evento interno “mov_IntE” el cual, a partir del movimiento configurado “mov_SP”, actualiza la posición de la cabeza “pos_SP”; *ii)* el evento de salida “outputSymbol_ExtEv”, el cual influencia la “tm_control”.

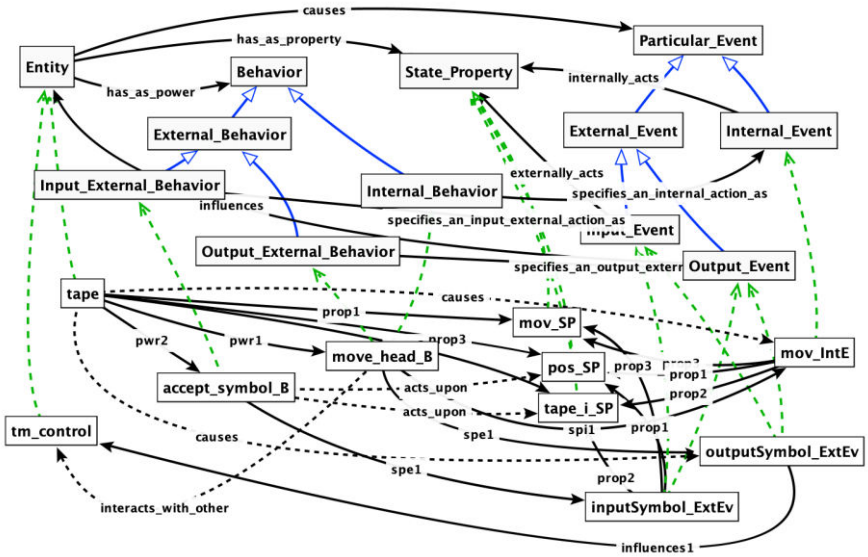


**Fig. 5.** Vista en ConceptBase de PEventCO en donde se instancia el “Tape System” propuesto en [7]: entidad tape.

La Figura 6 se centra en la representación del elemento “tm_control” que refiere a la entidad TM Control. Al igual que en la Figura 5, en la parte inferior de la figura se ilustran las instancias. En este caso, el evento “outputSymbol_ExtEv” especifica el comportamiento externo de entrada “query_transition_table_B”. Como parte del comportamiento externo de salida “apply_transition_B”, se genera el evento “inputSymbol_ExtEv”, el cual influencia a la entidad “tape”.

La Figura 7 incluye los resultados de las consultas de múltiples preguntas de competencia. En el recuadro superior izquierdo se incluye la respuesta a la pregunta de

competencia CQ2 (Cuáles son los eventos externos por los que reacciona la entidad) con el parámetro entidad igual a tape ("CQ2[tape/entity]"). De esta manera se consulta por los eventos externos que reacciona la entidad "tape". La instancia "inputSymbol_ExtEv" es instancia de "CQ2[tape/entity]", ya que es un "Input_Event" y actúa externamente sobre propiedades de "tape" (en la Figura 5, "inputSymbol_ExtEv" actúa sobre "pos_SP", "tape_i_SP", y "mov_SP").

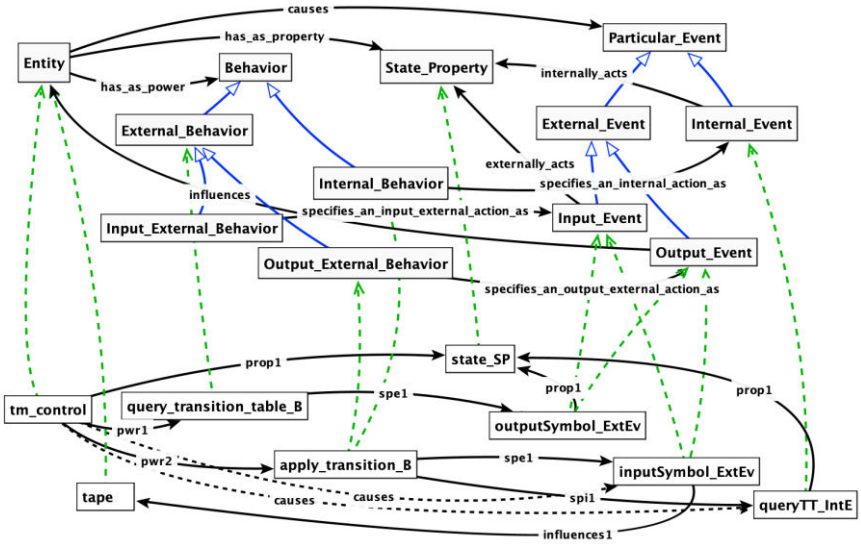


**Fig. 6.** Vista en ConceptBase de PEventCO en donde se instancia el "Tape System" propuesto en [7]: entidad tm_control.

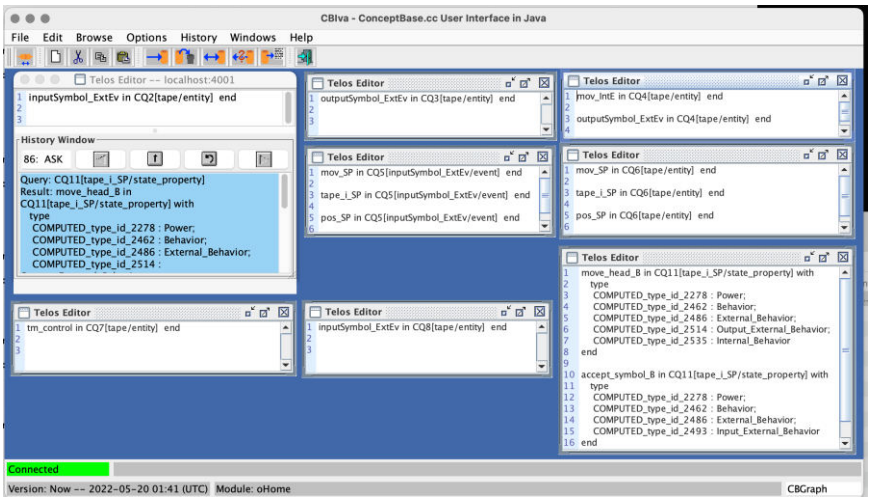


**Fig. 7.** Vista en ConceptBase de respuestas a las preguntas de competencia.

### 5.3 Discusión

A partir de la especificación de *PEventoCO* en Telos, ha sido factible especificar las distintas preguntas de competencia definidas en la Tabla 3. Asimismo, su implementación en ConceptBase facilitó la realización de casos de prueba, como el detallado en la subsección previa, y responder a las preguntas de competencia, obteniendo resultados esperados para cada una de ellas. Mediante un proceso iterativo e incremental, se fueron incorporando los axiomas necesarios y suficientes que permitieron responder correctamente a estas preguntas.

Luego, la ontología *PEventCO* ha dado soporte a la dinámica de comportamiento basada en eventos discretos requerida para el ejemplo del Tape System. La propuesta ha permitido representar los eventos que suceden en el fenómeno bajo estudio y los elementos estructurales que son afectados, instanciándose en este caso los eventos vinculados a la evolución del componente Tape de la máquina de Turing. Los axiomas propuestos establecen cierto nivel de integridad en *PEventCO*, así como entre el nivel CO y el nivel FO. Luego, esta prueba de concepto ha ilustrado que es posible la satisfacibilidad del modelo.

La principal diferencia que tiene esta propuesta con la ontología UFO-B [4], es que los eventos son conceptualizados como *Assertions*. En este caso, la semántica de dicho término definido a nivel fundacional enriquece la definición de los eventos permitiendo modelar mundos en los cuales la existencia de eventos se refleje a causa de la existencia de cosas particulares (en nuestro caso, entidades).

El modelo semántico propuesto en este trabajo es un primer paso hacia la definición de una representación para el nivel de "phenomenon" propuesto en [2]. En dicho trabajo, se propone una conceptualización basada en capas del formalismo de simulación Discrete-Event System Specification (DEVS) [7], donde la capa superior (denominada *phenomenon*) busca representar los distintos aspectos de un sistema dinámico de interés de forma independiente a la estrategia de simulación subyacente. En este contexto, la ontología *PEventCO* es un modelo semántico inicial que se ajusta a la especificación de sistemas discretos basados en eventos a partir de la modelización del mundo real.

## 6 Conclusiones y Trabajos Futuros

En este trabajo se ha presentado una ontología denominada *PEventCO* que facilita el modelado de entidades cuyos comportamientos dinámicos causan distintos tipos de eventos. Esta ontología de nivel *Core* se basa en la ontología fundacional *ThingFO*, por lo que sus términos, relaciones y propiedades reutilizan los elementos del nivel superior a fin de generar una compatibilidad dentro de la arquitectura FCD-OntoArch. Además, al situarse en dicho nivel, facilita la redefinición de sus conceptos en niveles de dominio como parte del trabajo futuro.

El principal beneficio de enmarcar la ontología de eventos dentro de la arquitectura FCD-OntoArch es que el contexto que rodea a las entidades que causan eventos y/o reaccionan a los mismos, puede ser definido en otros modelos ontológicos. Tales modelos pueden estar ubicados al mismo nivel que *PEventCO* o, en su defecto, a nivel inferior. Al mismo nivel, a priori se considera que la ontología *SituationCO* (ontología

de nivel CO para situaciones particulares y genéricas [20]) jugará un papel importante en relación con *PEventCO*. Cuando las entidades se contextualizan en una situación, es posible modelar una situación particular de un mundo particular. Sin embargo, una forma de enriquecer la descripción de dicha situación es por medio de la incorporación de los comportamientos asociados a las entidades involucradas. Luego, es factible pensar que sería posible definir situaciones donde las entidades participantes exhiban (o no) este tipo de comportamiento según el interés del modelador. Esto ayudará a enriquecer la semántica del mundo real para el caso de situaciones que son percibidas en base a comportamientos discretos.

## Referencias


1. Wainer, G. A.: Discrete-Event Modeling and Simulation: A Practitioner's Approach. CRC press (2017).
2. Blas, M., Gonnet, S., Zeigler, B. P.: Towards a Universal Representation of DEVS: A Metamodel-based Definition of DEVS Formal Specification. In Proceedings of the 2021 Annual Modeling and Simulation Conference (ANNSIM), 1-12 (2021). https://www.doi.org/10.23919/ANNSIM52504.2021.9552162.
3. Olivé, A., Raventós, R.: Modeling Events as Entities in Object-Oriented Conceptual Modeling Languages. Data & Knowledge Engineering, 58(3), 243-262 (2006). https://doi.org/10.1016/j.datak.2005.07.002
4. Guizzardi, G., Wagner, G., De Almeida Falbo, R., Guizzardi, R. S., Almeida, J. P. A.: Towards Ontological Foundations for the Conceptual Modeling of Events. In Proceedings of the 2013 International Conference on Conceptual Modeling, 327-341 (2013). https://doi.org/10.1007/978-3-642-41924-9_27
5. Olsina L.: Analyzing the Usefulness of ThingFO as a Foundational Ontology for Sciences. In Proceedings of the 2020 Argentine Symposium on Software Engineering, ASSE'20, 49 JAIIO, 172-191 (2020).
6. Blas, M., Gonnet, S., Tebes, G., Becker, P., Olsina, L.: Definición de una Ontología para la Representación de Comportamientos basados en Eventos. In Proceedings of the 2021 Argentine Symposium on Software Engineering, ASSE'21, 50 JAIIO, 38-51 (2021).
7. Zeigler, B. P., Muzy, A., Kofman, E.: Theory of Modeling and Simulation: Discrete Event & Iterative System Computational Foundations. Academic press (2018).
8. Guizzardi, G., Wagner, G.: Towards an Ontological Foundation of Discrete Event Simulation. In Proceedings of the 2010 Winter Simulation Conference, 652-664 (2010). https://doi.org/10.1109/WSC.2010.5679121
9. Gruber, T. A.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5(2), 199–220 (1993). https://doi.org/10.1006/knac.1993.1008
10. Wagner, G.: An Abstract State Machine Semantics for Discrete Event Simulation. In Proceedings of the 2017 Winter Simulation Conference, 762-773 (2017). https://doi.org/10.1109/WSC.2017.8247830
11. Pegden, C.D.: Advanced Tutorial: Overview of Simulation World Views. In Proceedings of the 2010 Winter Simulation Conference, 643−651 (2010). http://doi.org/10.1109/WSC.2010.5679161
12. Casati, R., Varzi, A.: Events, The Stanford Encyclopedia of Philosophy. Available at https://plato.stanford.edu/archives/sum2020/entries/events/ (2020).
13. Olsina L.: Applicability of a Foundational Ontology to Semantically Enrich the Core and Domain Ontologies. In proceedings of IC3K, KEOD'21, 13th International Conference on Knowledge Engineering and Ontology Development, held virtually in Oct. 2021, Portugal, pp. 1-9 (2021).

14. Guizzardi, G., Falbo, R., Guizzardi, R.: Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology. In *XI Conferencia Iberoamericana de Software Engineering (CIbSE'08)*, pp. 127-140 (2008)
15. Becker, P., Papa, M.F., Tebes, G., Olsina, L.: Analyzing a Process Core Ontology and its Usefulness for Different Domains. In Springer Nature Switzerland AG book, CCIS 1439: Int'l Conference on the Quality of Information and Communication Technology, A. C. R. Paiva et al. (Eds.): QUATIC'21, pp. 183-196 (2021).
16. Tebes, G., Olsina, L., Peppino, D., Becker, P.: Specifying and Analyzing a Software Testing Ontology at the Top-Domain Ontological Level. In: *Journal of Computer Science & Technology (JCS&T)*, vol. 21, no. 2, pp. 126-145 (2021). https://doi.org/10.24215/16666038.21.e12.
17. Fleetwood S.: The Ontology of Things, Properties and Powers. Journal of Critical Realism, 8:(3), 343-366 (2009).
18. ISO/IEC: International Organization for Standardization. *Information technology - Top-level ontologies (TLO) - Part 1: Requirements*. ISO/IEC Standard 21838-1, 1st Ed. 2021-08 (2021).
19. Koubarakis, M., Borgida, A., Constantopoulos, P. A retrospective on Telos as a meta-modeling language for requirements engineering. Requirements Eng 26, 1–23 (2021). https://doi.org/10.1007/s00766-020-00329-x
20. Olsina L., Tebes G., Becker P.: SituationCO v1.2's Terms, Properties and Relationships - A Core Ontology for Particular and Generic Situations. Preprint in Research Gate, Available at https://doi.org/10.13140/RG.2.2.23646.56644 (2021).

## Apéndice A: Preguntas de Competencia

La Tabla A1 presenta el listado de preguntas de competencia que ha sido detallado en la Tabla 3 como una traducción del idioma inglés.

**Tabla A1.** Preguntas de competencia usadas para definir el alcance de *PEventCO*.

| Category | Id. | Description |
|---|---|---|
| **Event** | **CQ1** | Which are the internal *events* that act over the state of an entity? |
| | **CQ2** | Which are the external *events* that make an entity react? |
| | **CQ3** | Which are the external *events* produced by an entity? |
| | **CQ4** | Which are the events caused by an entity? |
| **State Property** | **CQ5** | Which are the *state properties* related to the event? |
| | **CQ6** | Which are the *state properties* of the entity? |
| **Entity** | **CQ7** | Which are the *entities* influenced by the external events of the entity? |
| | **CQ8** | How is the entity influenced by external events? |
| | **CQ9** | Is the *entity* an input event consumer? |
| | **CQ10** | Is the *entity* an output event producer? |
| **Behavior** | **CQ11** | Which are the *behaviors* that act over a state property? |

## Apéndice B: Definiciones en inglés de PEventCO

Las Tablas B1 a B3 presentan las definiciones de los términos, propiedades, y relaciones no taxonómicas que han sido detalladas en las Tablas 4 a 6 como una traducción del idioma inglés.

**Tabla B1.** Definición de los términos definidos en *PEventCO*.

| Term | Definition |
|---|---|
| **Entity** | It represents a particular or concrete, tangible or intangible object, for which a dynamic behavior is defined explicitly using its Properties and Powers.<br>Note: Entity has the semantics of Thing –term coming from the ThingFO ontology. Therefore, an Entity is not a particular object without its Properties and Powers. |
| **Behavior** | It refers to what a particular Entity behaves under established conditions.<br>Note: Behavior has the semantics of Power –term coming from the ThingFO ontology. A Power is one member of the triad that conforms the unique individual named Thing. |
| **State Property** | It refers to the intrinsic state structure of a particular Dynamic Entity.<br>Note: State Property has the semantics of Property –term coming from the ThingFO ontology. A Property is one member of the triad that conforms to the unique individual named Thing. |
| **Particular Event** | It is an Assertion on Particulars and, at the same time, an Action-related Assertion that explicitly states and specifies the occurrence of an Entity action. It is related to the interaction and happening of Entities since acting Behaviors cause any Particular Events that might occur.<br>Note: Particular Event mechanisms need to consider Time Boundaries, in addition to the changes or queries in the states of the Entities' Properties. |
| **Time Boundary** | It is a Time-related Assertion that specifies temporal limits and restrictions from which a Particular Event or series of Particular Events can be related and modeled. |
| **Internal Behavior** | It is a type of Behavior that refers to what a particular Entity can do to act over its State Properties.<br>Note: It refers to the occurrence of internal actions on an Entity. |
| **External Behavior** | It is a type of Behavior that refers to the occurrence of external actions on an Entity. |
| **Input External Behavior** | It is a type of External Behavior that refers to the occurrence of external input actions over an Entity. |
| **Output External Behavior** | It is a type of External Behavior that refers to the occurrence of external output actions of an Entity. |

| | |
|---|---|
| **Internal Event** | It is a Particular Event that explicitly states and specifies the occurrence of an internal Entity action that acts over State Properties of such an Entity. |
| **External Event** | It is a Particular Event that explicitly states and specifies the occurrence of an external Entity action. |
| **Input Event** | It is an External Event that explicitly states and specifies the occurrence of an external Entity action that acts over State Properties of an Entity according to its External Behavior. |
| **Output Event** | It is an External Event that explicitly states and specifies the occurrence of an external Entity action that has some implication in other Entities. |

**Tabla B2.** Definición de las propiedades de los términos incluidos en *PEventCO*.

| Term | Attribute | Definition |
|---|---|---|
| **Particular Event** | **event specification** | It specifies a Particular Event. |
| **Time Boundary** | **time specification** | It specifies temporal relations and restrictions for Events since events happen in time. |

**Tabla B3.** Definición de las relaciones no taxonómicas de *PEventCO*.

| Relationship | Definition |
|---|---|
| **has as power** | An Entity has one or many Behaviors as Powers. |
| **has as property** | An Entity has one or many State Properties as Properties. |
| **causes** | A Dynamic Entity causes one or many Particular Events. |
| **specifies an internal action as** | An Internal Behavior specifies an Internal Event. |
| **specifies an input external action as** | An Input External Behavior specifies an Input Event. |
| **specifies an output external action as** | An Output External Behavior specifies an Output Event. |
| **internally acts** | An Internal Event internally acts on one or more State Properties. |
| **externally acts** | An Input Event externally acts on one or more State Properties. |
| **considers** | A Particular Event can be attached to Time Boundaries. |
| **influences** | An Output Event influences one or many Entities. |

## Apéndice C: The Tape System

El sistema *Tape System* se define como dos entidades independientes: *TM Control* y *Tape* (Figura C1). Es decir, que dicho sistema incluye la cinta (*Tape*) y su manejador (*TM Control*) el cual actúa como interfaz del controlador. El *Tape* es representado por una lista de símbolos que almacena información requerida por el *TM Control*. Esta

información únicamente puede ser accedida de a un símbolo por vez, haciendo uso de la posición actual del *head* (abreviada como *pos*).

Sobre el *Tape*, una interrupción externa que incluya un símbolo a hacer escrito (*sym*) y una instrucción de movimiento (*mv*), hace que se escriba dicho símbolo en la posición actual y, posteriormente, se actualice la posición *pos* a la indicada en el evento de entrada. Luego, se lee el símbolo en esta última posición y se lo envía como salida de la entidad. El *TM Control* obtiene dicho símbolo como una interrupción externa de entrada. A continuación, busca en su tabla interna (estado actual) el símbolo recibido como entrada a fin de generar el nuevo símbolo a ser enviado al *Tape* para que actualice su estado.

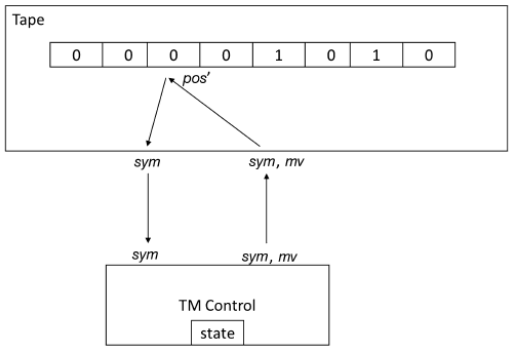


**Fig. C1.** Abstracción del Tape System como el sistema definido por el par (*Tape*, *TM Control*) (esquema tomado de [7]).