

Una representación basada en esquemas preconceptuales de eventos determinísticos y aleatorios tipo señal desde dominios de software científico

Paola Andrea Noreña Cardona, Carlos Mario Zapata Jaramillo

Universidad Nacional de Colombia, Medellín,
Colombia

{panorenac, cmzapata}@unal.edu.co

Resumen. Un evento representa un suceso que dispara un flujo de procesos. Una señal es un tipo de evento que inicia una alerta para cambiar un estado del sistema; estos eventos se clasifican en determinísticos y aleatorios. Determinísticos cuando su efecto se identifica con precisión y aleatorios cuando su causa no se puede predecir. Estos eventos se representan gráfica y matemáticamente en el modelado científico; en ingeniería de software se modelan gráficamente sin componentes científicos y los eventos no se clasifican en determinísticos y aleatorios. Por ello, hace falta un modelo en ingeniería de software que integre componentes científicos para representar eventos determinísticos y aleatorios tipo señal. En este artículo se propone mediante el esquema preconceptual; este esquema es un modelo gráfico de ingeniería de software para representar un dominio. La representación propuesta permite a los analistas representar dominios de software científico que incluyan este tipo de eventos.

Palabras clave: dominios de software científico, representación de eventos, evento determinístico, evento aleatorio, señal, esquemas preconceptuales.

Deterministic/Random Signal-Type Events Representation from Scientific Software Domains based on Pre-conceptual Schemas

Abstract. An event represents an occurrence, which triggers a process flow. A signal is an event type starting an alert for changing a state of the system; such events are classified as deterministic and random. Deterministic has an effect precisely identified and random produces an unpredictable cause. Such events are graphically and mathematically represented in scientific modeling; also, they are graphically modeled in software engineering, but such models lack scientific components; also, they lack the deterministic and random event classification. Thus, software engineering lacks a model with scientific components for representing deterministic/random signal-type events. In this paper, we propose a representation about such events from scientific software domains by using pre-conceptual schemas; such schemas are software engineering graphical models for

representing a domain. Such representation is proposed for allowing analysts to represent scientific software domains by including such event types.

Keywords: scientific software domain, event representation, deterministic event, random event, signal, pre-conceptual schemas.

1. Introducción

Los eventos representan sucesos que, al cumplir con condiciones y restricciones, disparan el flujo de procesos y propician cambios de estado en el comportamiento del sistema [1,2]. Los eventos tipo señal aparecen en el sistema para generar una alerta de un cambio de estado en las variables del sistema [3]. Estos eventos establecen la comunicación entre un emisor que transmite la señal y un receptor que la recibe [4]. Estas señales pueden ser eventos determinísticos o aleatorios; el término *determinístico* permite describir eventos a partir de ecuaciones matemáticas y reglas o condiciones; los eventos determinísticos tipo señal incluyen un valor resultante (efecto) [5]. El término *aleatorio* permite describir un conjunto de datos no determinísticos; los eventos aleatorios tipo señal no se pueden predecir con precisión y, generalmente, se describen mediante variables aleatorias, ya que se desconoce el momento en que van a suceder [6]. Ambos tipos de eventos surgen producto de la ocurrencia de otros eventos [5].

Muñoz *et al.* [7], de las Morenas *et al.* [8] y Ramírez y Colina [9] representan eventos determinísticos tipo señal y Priyadharshini y Justin [10] y Di Leo *et al.* [11] representan eventos aleatorios tipo señal mediante modelos gráficos y matemáticos del modelado científico. En los anteriores enfoques no se agrupa la representación matemática de los eventos en la representación gráfica. Pascual *et al.* [12] modelan matemáticamente eventos aleatorios. Armendáriz [6] propone un enfoque que integra los eventos determinísticos y aleatorios tipo señal en un modelo científico. En el OMG (Object Management Group) [13–14] se propone la representación gráfica de eventos tipo señal en ingeniería de software, sin diferenciar su clasificación científica. Aunque Bazydlo *et al.* [15] y Armas *et al.* [16] describen una traducción de modelos de ingeniería de software a modelos científicos, tampoco se enfocan en su clasificación.

Por lo anterior, la ingeniería de software carece de un modelo que integre componentes científicos (gráficos y matemáticos) para representar eventos determinísticos y aleatorios tipo señal desde dominios de software científico (estos dominios involucran las diferentes áreas científicas como ciencias, matemáticas, ingenierías y medicina, entre otros [17]). Adicionalmente, el modelado científico carece de un modelo que integre componentes de ingeniería de software en sistemas que incluyen estos eventos.

En este artículo se propone una representación de eventos determinísticos y aleatorios tipo señal en dominios de software científico a partir de esquemas preconceptuales (EP). Esta representación permite una vista de sistemas que operan con señales como el sistema de telefonía celular. El EP es un modelo de ingeniería de software para representar el dominio de un sistema a partir de componentes dinámicos (que permiten representar la parte comportamental del sistema) y estructurales (que permiten representar las relaciones entre conceptos). Estos componentes le dan una vista completa del dominio a los analistas y a los interesados del sistema [18].

La representación de eventos determinísticos y aleatorios tipo señal en el EP integra componentes con los que se representan estos eventos en el modelado científico. La integración de estos componentes permite que la ingeniería de software tenga un modelo para representar eventos en sistemas de software científico.

Este artículo se estructura de la siguiente manera: en la Sección 2 se presenta el marco teórico; en la Sección 3 se exponen los antecedentes; en la Sección 4 se plantea el problema, en la Sección 5 se propone la solución. Finalmente, se presentan las conclusiones y el trabajo futuro.

2. Marco teórico

2.1. Eventos

Los eventos representan sucesos o acontecimientos de algo significativo en el flujo de procesos que, al cumplir con condiciones y restricciones, disparan el comienzo o el fin de una operación. Los eventos son responsables del cambio de estados en el comportamiento del sistema; así, la información que se obtiene a partir de los eventos permite analizar este comportamiento. Por ello, la representación de los eventos se vuelve necesaria en la ingeniería de software [1, 2]. El evento que dispara el comienzo de un proceso, un flujo de procesos u otro evento se conoce como *evento disparador* [19]. El evento que surge como resultado del fin de un proceso o un flujo de procesos se conoce como evento de resultado [19, 20].

Eventos tipo señal: Estos eventos constituyen eventos disparadores que, mediante una alerta, cambian el estado de las variables del sistema [3]. Según la notación del modelado de procesos de negocio (BPMN, por sus siglas en inglés), los eventos tipo señal se utilizan para representar señales en un sistema de comunicación, donde un transmisor envía una señal y un receptor la recibe [4]. En dominios científicos, como la electrónica, estas señales se clasifican en eventos determinísticos y eventos aleatorios, que pueden surgir producto de otros eventos. De acuerdo con esta clasificación, los analistas científicos pueden modelar estas señales a partir de modelos gráficos y matemáticos [5].

Evento determinístico tipo señal: Se representan con ecuaciones matemáticas y reglas o condiciones. Estos eventos implican un valor resultante que es predecible, debido a que la relación causa-efecto se conoce en su totalidad durante la ocurrencia del evento [5].

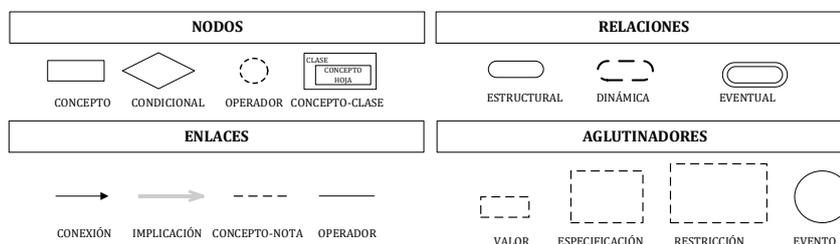


Fig. 1. Elementos de la notación del EP [18].

Evento aleatorio tipo señal: Se representa con variables aleatorias o ecuaciones matemáticas con información variable que no puede ser determinística, ya que no se conciben valores predecibles, por lo que se describe un conjunto de valores dentro de los resultados posibles que representan una posible causa [6]. Un evento puede ser aleatorio en su causa y su efecto puede generar un evento determinístico. Por ejemplo, cuando una persona pasa por un sensor en un instante impredecible (evento aleatorio) y genera que la luz se encienda (evento determinístico); ambos eventos requieren el tiempo para identificar su ocurrencia.

2.2. Esquema preconceptual (EP)

El EP es un modelo de ingeniería de software para representar computacionalmente un dominio a partir de componentes estructurales y dinámicos. Estos componentes se incluyen para solventar la vista completa del dominio y facilitar su comprensión gráfica. Además, el EP se utiliza como modelo base en UNC-Method (Método de educación de requisitos de la Universidad Nacional de Colombia) para relacionar todos los productos de trabajo en el proceso de ingeniería de software [18].

Los analistas de negocios pueden utilizar la notación gráfica del EP para representar eventos y demás elementos de un dominio mediante herramientas de modelado como Visio de Microsoft™ y Draw.io de Google (gratuita). Los elementos de la notación del EP se basan en reglas lingüísticas (véase la Fig. 1): los *nodos* pueden ser *concepto* (por ejemplo, celular), *condicional* (por ejemplo, si estado de llamada = “conectada”), *operador* (operadores lógicos: *and*, *or* y aritméticos: **/*, *+*, *-*) y *concepto-clase* (por ejemplo, “llamada” es concepto clase y “estado” es su concepto hoja); las *relaciones* pueden ser *estructural* (verbo de dependencia *tiene*), *dinámica* (verbo de acción; por ejemplo, persona guarda informe; también se utilizan en procesos automáticos; por ejemplo, actualiza potencia del celular) y *eventual* (verbo de evento, que no requiere un sujeto; por ejemplo, tiempo pasa); los *enlaces* pueden ser *conexión* (enlace entre un concepto y una relación; por ejemplo, alerta→emerge), *implicación* (enlace causa-efecto; por ejemplo, inserta potencia→inserta estación), *concepto-nota* (enlace valor, restricción y especificación) y *operador*; y los *aglutinadores* pueden ser *valor* (por ejemplo, ‘2’), *especificación* (por ejemplo, estado de llamada= “desconectada”), *restricción* (por ejemplo, si potencia de estación >= potencia de celular→inserta potencia del celular) y *evento* (agrupa concepto y verbo de un evento; por ejemplo llamada inicia).

2.3. Dominios de software científico

Los dominios de software científico representan el conocimiento de sistemas de software, que incluyen información específica acerca de contextos científicos. Estos dominios contienen información en áreas de ingeniería, medicina, ciencias, matemáticas, administración y economía. Los sistemas de software en estos dominios se utilizan en la resolución de problemas y descripción de fenómenos naturales, biológicos o mecánicos. Estos sistemas se clasifican en *software industrial* (sistemas de software con procesos complejos que expertos científicos apoyan), *software de investigación* (sistemas de software que respaldan una investigación específica en un

dominio científico) y *software a pequeña escala* (sistemas de software que desarrollan investigadores o estudiantes en las diferentes áreas científicas) [17].

3. Antecedentes

En algunos enfoques del modelado científico se incluye la representación de eventos determinísticos tipo señal mediante modelos gráficos y matemáticos. Muñoz *et al.* [7] presentan una red de Petri y un modelo matemático para representar la combinación de señales de entrada y salida. Sin embargo, no incluyen el modelo matemático en la red de Petri. De las Morenas *et al.* [8] implementan un control de un centro de distribución automatizado mediante agentes físicos y dispositivos de señales de radiofrecuencia RFID; ellos representan estas señales mediante redes de Petri, pero no presentan estructuras matemáticas para ellas. Ramírez y Colina [9] incluyen eventos determinísticos tipo señal en una red neuronal para sensores, los cuales verifican la presencia de señales fuera del rango de operación; estas señales indican fallos en la cantidad y calidad de la producción de los pozos de petróleo. La red neuronal se basa en un modelo matemático, pero no se integra a la red.

En otros enfoques del modelado científico se incluye la representación de eventos aleatorios tipo señal mediante modelos gráficos y matemáticos. Priyadharshini y Justin [10] representan los eventos aleatorios tipo señal en un diagrama de bloque; estos eventos son señales de energía en pacientes con válvula mecánica cardíaca y estenosis pulmonar; aunque los eventos también se representan mediante ecuaciones matemáticas, no se integran al diagrama de bloque. Di Leo *et al.* [11] presentan una simulación del modelo matemático en la detección de cambios en señales aleatorias de estructuras turbulentas en un flujo de aire; también, Pascual *et al.* [12] representan eventos aleatorios tipo señal con un modelo matemático en procesos Gaussianos para la varianza del ruido en sensores. Sin embargo, estos enfoques no tienen una representación gráfica de los modelos matemáticos.

Por otra parte, Armendáriz [6] incluye eventos determinísticos y aleatorios tipo señal mediante un autómata finito y modelos matemáticos para detectar fallos, por variación de la frecuencia en la rotación de la hélice de un avión; a pesar de ello, no se integran las estructuras de los modelos matemáticos en el autómata finito. Los anteriores enfoques carecen de componentes de ingeniería de software.

En algunos enfoques del modelado en ingeniería de software se proponen modelos gráficos de eventos tipo señal: En OMG se proponen estructuras gráficas en el modelo de procesos BPMN [13] y en el diagrama de máquina de estados [14] del lenguaje de modelado unificado (UML por sus siglas en inglés); ambos enfoques son modelos tradicionales que sólo permiten presentar una vista dinámica o estructural de un sistema de software, los cuales carecen de la integración de estructuras matemáticas para representar eventos determinísticos y aleatorios. En otros enfoques de este modelado, se describe una traducción de diagramas de ingeniería de software al modelado científico: Bazydlo *et al.* [15] proponen un diagrama de máquina de estados UML que incluye eventos tipo señal, exportan el diagrama al lenguaje de marcado extensible (XML por sus siglas en inglés) y lo traducen al modelo de máquina de estados finitos jerárquicos (HCFSM por sus siglas en inglés) para su posterior simulación; Armas *et al.* [16] plantean una traducción de eventos desde un modelo de procesos BPMN a una

red de Petri. Sin embargo, ninguno de estos enfoques contempla componentes científicos como las estructuras matemáticas para la representación de eventos en la traducción, ni se enfocan en la clasificación científica de los eventos determinísticos y aleatorios.

4. Planteamiento del problema

Según los enfoques de la Sección 2, usualmente en el modelado científico dos modelos acompañan la representación de eventos determinísticos y aleatorios tipo señal: el modelo gráfico y el modelo matemático; esto permite inferir que ambos modelos se requieren en la representación de estos eventos. A diferencia de este modelado, en ingeniería de software se utilizan sólo modelos gráficos para representar eventos tipo señal y no se enfocan en la clasificación científica de eventos determinísticos y aleatorios, es decir que hacen falta componentes científicos. Además, los modelos tradicionales sólo presentan una vista del sistema (dinámica o estructural). Por ello, la ingeniería de software carece de un modelo que integre componentes científicos (gráficos y matemáticos) para representar eventos determinísticos y aleatorios tipo señal en dominios de software científico. Adicionalmente, se requiere este modelo porque los analistas científicos no integran procesos de ingeniería de software en el desarrollo de software [21,22] y los analistas de negocios (expertos en prácticas de ingeniería de software) no tienen el conocimiento y los componentes para entender estos dominios y producir software científico.

5. Propuesta de solución

La propuesta de solución se basa en los EP como modelos para la representación de los eventos, ya que en los EP se integran componentes de software y de lingüística computacional a diferencia de los demás modelos como: i) componentes estructurales y dinámicos en un mismo modelo para representar la vista completa del sistema [18]; ii) estructuras lingüísticas, gráficas y matemáticas para representar eventos disparadores [18,20]; esto permite la integración de estructuras matemáticas y demás componentes que proceden de modelos matemáticos para representar eventos determinísticos y aleatorios tipo señal en dominios de software científico.

5.1. Integración de componentes científicos para representar eventos determinísticos y aleatorios tipo señal

Las ecuaciones matemáticas son “autocontenidas”, es decir, integran elementos de un contexto en la operación; para entender estos elementos se requiere conocer previamente la documentación del contexto en el cual se aplican. Mediante la solución propuesta se pretende integrar las estructuras matemáticas con los elementos del contexto que permitan entender el comportamiento de los eventos determinísticos y aleatorios tipo señal en el mismo modelo a partir de la notación del esquema preconceptual. Para ello, esta fase se divide en dos etapas:

Traducción de elementos de una ecuación matemática a su forma conceptual:

Los eventos determinísticos se suelen representar con ecuaciones matemáticas que permiten dar un resultado predecible [24]. Para los eventos tipo señal, las ecuaciones representan el efecto que causa la señal; por ejemplo, al observar la ecuación (1): i) se debe indicar el contexto en el cual se fórmula, es decir, la ecuación de la potencia que recibe un celular desde una estación transmisora en la red de telefonía celular y ii) se deben identificar los elementos que se encuentran autocontenidos, es decir, Pr es la potencia de señal recibida, Pt es la potencia transmitida de la estación, cte es la constante de propagación de la señal, d es la distancia entre la estación (transmisor) y el celular (receptor) y n es el coeficiente de propagación del entorno, que en valores urbanos puede tomar valores entre 2,5 y 5 W (Watts) [25]:

$$Pr = \frac{Pt}{cte} * d^n. \quad (1)$$

Para traducir esta ecuación al evento tipo señal *señal aparece* en el celular, se utiliza una regla de los EPs que exige que los elementos que se describen en su notación deben ser conceptos con palabras completas que faciliten la comprensión del dominio. De esta manera, se utiliza el nombre de cada variable y parámetro (constante) de la ecuación. Al tomar el mismo ejemplo, la ecuación (1) se presenta en su forma conceptual en la ecuación (2).

$$Potencia\ recibida = \frac{Potencia\ transmitida}{Constante\ de\ transmisión} * distancia^{coeficiente\ de\ propagación\ del\ entorno}. \quad (2)$$

Los eventos aleatorios se suelen representar mediante ecuaciones con variables aleatorias que dan una probabilidad al valor. Estos valores también se utilizan para efectuar simulación de sistemas. Para los eventos tipo señal se relaciona al momento impredecible en que ocurre la señal. La función *random* en las plataformas de desarrollo de software se integran para generar valores aleatorios entre 0 y 1 y generalmente se presenta como *rand()*. Para aplicar esta función se asigna a la variable aleatoria y se traduce como se presenta en la ecuación (3). Esta función puede variar según el requisito. Si se requiere que el valor aleatorio se encuentre entre dos valores, se utiliza la ecuación (4), donde X es el valor aleatorio, m es el límite inferior y n es el límite superior. De esta manera, se traduce en la ecuación (5); si el requisito es para dos valores, se utiliza la función *rand* con ambos valores numéricos o valores *string* (caracteres):

$$Variable\ aleatoria = rand, \quad (3)$$

$$X = m + rand() \% n + 1 - m, \quad (4)$$

$$Variable\ aleatoria = límite\ inferior + rand \% límite\ superior + 1 - límite\ inferior. \quad (5)$$

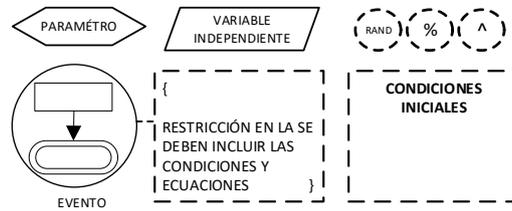


Fig. 2. Estructuras para representar eventos determinísticos y aleatorios tipo señal.

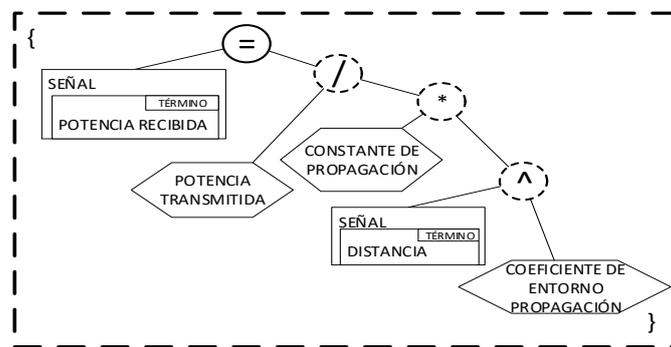


Fig. 3. Representación de la ecuación de la potencia transmitida.

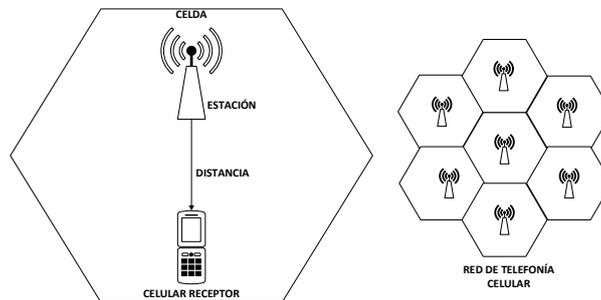


Fig. 4. Sistema de telefonía celular a partir del celular receptor, elaboración propia a partir de [25].

Definición de estructuras para representar eventos determinísticos y aleatorios tipo señal: Se aplica la notación de las restricciones del EP para definir estructuras que permitan la representación de eventos determinísticos y aleatorios tipo señal, como condiciones y ecuaciones matemáticas desde dominios de software científico a partir de la traducción en la etapa anterior. Para las ecuaciones matemáticas se utilizan elementos gráficos de la notación del esquema, como conceptos, operadores y conectores de operadores, y se integran otras estructuras como el *parámetro*, la *variable independiente* para definir variables globales como el tiempo; algunos operadores como el *operador rand*, el *porcentaje (%)* para generar variables aleatorias y el *operador exponencial (^)* con base en un valor y las *condiciones iniciales* que se utilizan en dominios de software científico para simular los sistemas (véase la Fig. 2).

Las ecuaciones matemáticas se elaboran a partir de árboles binarios, que comúnmente se usan para representar expresiones algebraicas y booleanas [23]. Por ejemplo, para representar la ecuación (2), se incorporan las estructuras anteriores en una *restricción* que se relaciona al evento (véase Fig. 2) surgiendo la representación de la Fig. 3. Estas restricciones también se utilizan para condiciones del evento.

5.2. Una representación basada en esquemas preconceptuales de eventos determinísticos tipo señal desde dominios de software científico

La integración de componentes científicos en el EP se aplica al dominio científico de electrónica en un sistema de red de telefonía celular que, junto con los componentes de software del EP permiten una vista dinámica y estructural del sistema. El nivel de detalle se centra en el funcionamiento del sistema para el celular que recibe una llamada; así, la representación de eventos se orienta a eventos determinísticos y aleatorios tipo señal, que suceden en el celular receptor y las características que se modelan se basan en su ocurrencia respecto del tiempo.

En la Fig. 4 se puede apreciar una red de telefonía celular que se distribuye en celdas, las cuales constituyen el área de cobertura de una estación o antena. Estas estaciones funcionan como transmisores de potencia, que son las que permiten que un celular reciba la potencia de la señal, es así como un celular que hace parte de la red puede recibir una llamada. La señal de los celulares debe ser baja para que otros móviles pueda reutilizar los canales de las celdas [25].

En la Fig. 5 se elabora un EP del dominio del sistema utilizando las estructuras propuestas.

El EP se representa con la funcionalidad automática que integran los simuladores de sistemas de eventos. Este modelo se logró a partir del conocimiento de un experto y fuentes en el área.

Los componentes científicos se aprecian en el EP comenzando en las *condiciones iniciales*, las cuales permiten ejecutar la vista dinámica del sistema con valores iniciales de parámetros y variables globales e internas. La secuencia de eventos inicia con el evento *tiempo pasa*, el cual se representa con una restricción que incluye un ciclo que inicia en '0' según la condición inicial *término=0*, y termina en '1440' según la condición *término < 1440*. *Término* es el contador de vectores que permite controlar la posición del tiempo y el registro en las ocurrencias de cada evento; el valor *1440* identifica las iteraciones por el *tiempo 1440 minutos* que equivalen a un 1 día, el cual incrementa de por cada '*1 minuto*'. Este evento controla la secuencia de los demás eventos del sistema, ya que cada ocurrencia se toma respecto del tiempo.

El evento *señal aparece* representa la señal que el *celular receptor* tiene a partir de la *estación* transmisora. Este evento tipo señal es determinístico, ya que se puede representar mediante la condición si *potencia recibida[término]* de la *señal < potencia recibida[término-1]* (que permite al celular buscar la señal más alta de las estaciones de transmisión) y la ecuación (2), dando un resultado predecible de la potencia recibida, que se deriva de los elementos de la fórmula y las condiciones iniciales *potencia transmitida=5 W* (valor por normatividad que deben emitir las estaciones transmisoras y que puede variar según la regulación de un país), *la constante de propagación=2 W/m* y el *coeficiente de entorno propagación=3 W* (puede variar según el entorno urbano o rural, ya que el entorno urbano requiere más celdas en el sistema).

La especificación de este evento contiene procesos automáticos que se ejecutan para guardar el valor de la *potencia recibida*; de esta manera, el celular constantemente guarda el registro en un instante de tiempo, como se observa en la relación dinámica *inserta tiempo local[término]*, luego *inserta el código* de la *estación* a la que se está conectando el celular receptor y la *distancia* entre esta estación y el celular. La *distancia* hace que la señal varíe, por lo cual, si es menor que la anterior se *actualiza código estación*, se *actualiza distancia[término]*, para finalmente *actualizar potencia recibida[término]*.

Tabla 1. Funcionamiento del evento *tiempo pasa*.

TIEMPO PASA	
TÉRMINO	TIEMPO
0	0 minutos
1	1 minuto
120	120 minutos
1440	1440 minutos

Tabla 2. Funcionamiento del evento *llamada llega*.

LLAMADA LLEGA		
MOMENTO	VARIABLE ALEATORIA	ALARMA
0 minutos	0,0	INACTIVA
1 minuto	0,6	INACTIVA
260 minutos	0,7	ACTIVA
1440 minutos	0,8	ACTIVA

Tabla 3. Tabla del concepto-clase *celular receptor*.

CELULAR RECEPTOR	
NÚMERO	PROPIETARIO
3002342111	Paola Noreña

La *llamada llega* es un evento aleatorio, el cual se representa a partir de una condición que guarda el instante (impredecible) en el que puede ocurrir el evento; si esto ocurre se activa la llamada. La *llamada llega* no implica que se inicie una llamada, ya que el receptor puede aceptar o rechazar la llamada, el evento *llamada inicia* surge a partir de la respuesta del receptor; sin embargo, no es de interés para los expertos del dominio de electrónica conocer la persona o rol que ejecuta la acción, sino que se enfoca en el funcionamiento de los elementos del sistema. Es así como, en la representación de este evento, se incluye la respuesta aleatoria, que puede ser “*aceptada*” o “*rechazada*”, lo que requiere conocer la comunicación que se establece en el sistema al ser “*aceptada*”, inserta el *estado* de la llamada “*conectada*” y el *tiempo inicial*; luego surge el evento *llamada finaliza* a partir del *estado* de la llamada “*desconectada*”, con el cual se inserta *tiempo final* y *duración*.

La vista estructural del sistema en el EP, permite representar la información que se deriva del sistema a partir de los eventos y los conceptos clase son *estación* (cuyos

Tabla 4. Tabla del concepto-clase *celular emisor*.

CELULAR EMISOR	
NÚMERO	PROPIETARIO
3203234556	Luis Fernando Guarín
3142342112	Martha Lucía Cardona

Tabla 5. Tabla del concepto-clase *estación*.

ESTACIÓN	
CÓDIGO	CELDA
123.12.21.1	Robledo
123.12.21.3	Laureles

Tabla 6. Tabla del funcionamiento del evento *señal aparece*.

SEÑAL APARECE				
TÉRMINO	TIEMPO LOCAL	CÓDIGO DE ESTACIÓN	DISTANCIA	POTENCIA RECIBIDA
0	0 minutos	120.34.23.1	100 m	2 W
1	1 minuto	120.34.23.1	1000 m	3 W
2	2 minutos	120.34.23.1	150 m	4 W

Tabla 7. Tabla del concepto-clase *llamada*.

LLAMADA						
NUMERO CELULAR EMISOR	NÚMERO CELULAR RECEPTOR	RESPUESTA	ESTADO	TIEMPO INICIAL	TIEMPO FINAL	DURACIÓN
3203234556	3002342111	Aceptada	Desconectada	120 minutos	180 minutos	60 minutos
3142342112	3002342111	Rechazada	Desconectada	220 minutos	220 minutos	0 minutos
3142342112	3002342111	Aceptada	Desconectada	430 minutos	460 minutos	30 minutos

conceptos hoja o atributos son *celda*, *ubicación* y *potencia*), *celular receptor* (cuyos conceptos hoja son *número* y *propietario*), *celular emisor* (cuyos conceptos hoja son *número* y *propietario*), *estación* (cuyos conceptos hoja son *código* y *celda*), *señal* (cuyos conceptos hoja son *tiempo local*, *distancia* y *potencia recibida*) y *llamada* (cuyos conceptos hoja son *respuesta—aceptada o rechazada*), *estado* (*conectada*, *desconectada*), *tiempo inicial*, *tiempo final* y *duración*).

Mediante las siguientes tablas de datos se puede verificar el funcionamiento de la vista estructural del EP y de la base de datos del sistema de software.

6. Conclusiones y trabajo futuro

La representación que se propuso en este artículo basada en el EP permite modelar eventos determinísticos y aleatorios tipo señal en dominios de software científico, en una vista completa del dominio que integra la vista estructural y la vista dinámica. Esta representación incluye elementos nuevos al EP, como las condiciones iniciales, los

parámetros y las variables independientes, además de una serie de operadores comunes a la notación matemática y que previamente no se incluían en el EP. La representación de estos eventos en el EP constituye un enfoque que permite unificar componentes científicos y componentes de software en un mismo modelo de ingeniería de software para la producción de software científico.

Analistas científicos y analistas de negocios pueden utilizar el EP para representar y comprender el funcionamiento de eventos determinísticos y aleatorios tipo señal en dominios de software científico.

Como trabajo futuro se pueden adicionar otras estructuras complejas para representar otro tipo de eventos y elementos de los dominios de software científico, como ecuaciones diferenciales o análisis estadísticos.

Agradecimientos. Este artículo es producto del proyecto de investigación Doctoral *una extensión a esquemas preconceptuales para el refinamiento en la representación de eventos y la notación matemática* con código Hermes 39886 de la Universidad Nacional de Colombia, que financia Colciencias en la convocatoria 727 de becas para estudiantes de Doctorado en Colombia.

Referencias

1. Zapata, C.M., Noreña, P.A., Gonzales, N.: Representación de eventos disparadores y de resultado en el grafo de interacción de eventos. Ing. UsbMed, 4(2), pp. 23–32 (2013)
2. Noreña, P.A., Zapata, C.M.: A Game for Learning Event-Driven Architecture: Pre-conceptual-Schema-based Pedagogical Strategy. Development in Business Simulation and Experiential Learning, 45, pp. 312–319 (2018)
3. Noreña, P.A., Vargas, F.A., Soto, D.E.: Tipificación de eventos a partir del modelo BPMN en artefactos de ingeniería de software. Cuaderno Activa, 6, pp. 49–61 (2014)
4. Von-Rosing, M., Von-Scheel, H., Scheer, A.W.: The Complete Business Process Handbook: Body of Knowledge from Process Modeling to BPM. Morgan Kaufmann, 1 (2014)
5. Ruhm, K.H.: Deterministic, Nondeterministic Signals. Internet Portal Measurement Science and Technology. <http://www.mmm.ethz.ch/dok01/d0000839.pdf> (2008)
6. Armendariz, I.: Análisis paramétrico determinista y aleatorio de problemas dinámicos en estructuras aeroespaciales. Tesis Doctoral en Aeronáutica, Universidad Politécnica de Madrid (2016)
7. Muñoz, D.M., Correcher, A., García, E., Morant, F.: Generación determinística de lenguajes legales para sistemas de eventos discretos. Revista Iberoamericana de Automática e Informática Industrial, (RIAI), 13(2), pp. 207–219 (2016)
8. De las Morenas, J., García, A., Martínez, F., Ansola, P.G.: Implementación del control en planta de un centro de distribución automatizado mediante agentes físicos y RFID. Revista Iberoamericana de Automática e Informática Industrial, (RIAI), 12(1), pp. 25–35 (2015)
9. Ramírez, M., Colina, E.: Sistema supervisor inteligente para procesos de producción de petróleo. Revista científica Maskana, en Congreso (MATCH'14), pp. 71–82 (2016)
10. Priyadharshini, V., Justin, J.: Quantification of Non-deterministic Events in Pathological PCG signals using Continuous and Packet Wavelet Transforms. In: International Conference on Communications and Signal Processing (ICCSP'14), pp. 1419–1423 (2014)
11. Di Leo, J.M., Calandra, M.V., Delnero, J.S.: Algoritmos de punto de cambio aplicados a la detección de estructuras vorticosas en flujos turbulentos. Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería, 33(3), pp. 225–234 (2017)

12. Pascual, J.P., von-Ellenrieder, N., Muravchik, C.: Cramér-Rao Bound for Parameter Estimation in Sensor Arrays with Mutual Coupling. *IEEE Latin America Transactions*, 11(1), pp. 91–96 (2013)
13. OMG, Object Management Group: Business Process Model and Notation BPMN. <http://www.omg.org/spec/BPMN/1.2> (2014)
14. OMG, Object Management Group: Superstructure 2.5. <http://www.omg.org/spec/UML/2.5/> (2015)
15. Bazydło, G., Adamski, M., Stefanowicz, L.: Translation UML diagrams into Verilog. In: 7th International Conference on Human System Interactions (HIS'14), pp. 267–271 (2014)
16. Armas, A., Baldan, P., Dumas, M., Garcia, L.: Diagnosing behavioral differences between business process models: An approach based on event structures. *Information Systems*, 56, pp. 304–325 (2016)
17. Kelly, D.: Scientific software development viewed as knowledge acquisition: Towards understanding the development of risk-averse scientific software. *Journal of Systems and Software*, 109, pp. 50–61 (2015)
18. Zapata, C.M.: The UNC-Method revisited: elements of the new approach. Saarbrücken: Lambert (2012)
19. Zapata, C.M., Noreña, P.A., Vargas, F.A.: The Event Interaction Game: Understanding Events in the Software Development Context. *Developments for Business Simulation and Experiential Learning*, 41, pp. 256–263 (2014)
20. Noreña, P.A.: Un mecanismo de consistencia en los eventos disparador y de resultado para los artefactos de UNC-Method. Tesis de Maestría, Universidad Nacional de Colombia (2013)
21. Wilson, G., Aruliah, D.A., Brown, C.T., Chue-Hong, N.P., Davis, M., Guy, T., Steven-Haddock, H.D., Huff, D., Mitchell, I.M., Plumbley, M.D., Waugh, B., White, E.P., Wilson P.: Best Practices for Scientific Computing. *PLOS Biology*, 12(1), pp. 1–6 (2014)
22. Kanewala, U., Bieman, J.M.: Testing scientific software: A systematic literature review Author links open overlay. *Information and Software Technology*, 56(10), pp. 1219–1232 (2014)
23. Leon, A.: Probability, Statistics, and Random Processes for Electrical Engineering. Pearson Prentice Hall (2017)
24. Kuipers, J., Ueda, T., Vermaseren, J.A.M.: Code optimization in FORM. *Computer Physics Communications*, 189, pp. 1–19 (2015)
25. Cruz-Ormetta, V.: Telefonía móvil y su salud. Lima: INICTEL (2005)