

# Un Editor Textual para el Modelado y la Generación de Código de Patrones de Eventos

Juan Boubeta-Puig<sup>1</sup>, Ramón Ramírez-González<sup>2</sup>, Guadalupe Ortiz<sup>1</sup>

<sup>1</sup>Departamento de Ingeniería Informática, Escuela Superior de Ingeniería, Universidad de Cádiz, Avda. de la Universidad de Cádiz 10, 11519 Puerto Real, Cádiz, España  
{juan.boubeta, guadalupe.ortiz}@uca.es

<sup>2</sup>Escuela Superior de Ingeniería, Universidad de Cádiz, Avda. de la Universidad de Cádiz 10, 11519 Puerto Real, Cádiz, España  
rrgonzalez1992@gmail.com

**Resumen.** El procesamiento de eventos complejos (CEP) es una tecnología que permite analizar y correlacionar grandes cantidades de datos con el propósito de detectar situaciones de interés en tiempo real. Para ello se requiere implementar patrones de eventos, especificando las condiciones que deben cumplirse para detectar dichas situaciones, con los lenguajes de procesamiento de eventos (EPL). A pesar de que los usuarios suelen tener un vasto conocimiento en el dominio para el que se necesitan definir ciertos patrones de eventos, suelen ser inexpertos tanto en EPL como en el lenguaje requerido para implementar las acciones a llevar a cabo tras la detección de los patrones. En este artículo presentamos un editor textual para el modelado y la generación de código de los patrones de eventos que se necesiten detectar en un dominio de aplicación. Gracias a este editor, el usuario solo tendrá que conocer un lenguaje textual para definir patrones de eventos, que podrán ser posteriormente transformados automáticamente al EPL soportado por el motor CEP en cuestión. Este editor complementa a MEdit4CEP, un editor que permite la definición gráfica e intuitiva de patrones sin necesidad de conocer ningún lenguaje de programación en particular.

**Keywords:** CEP, EPL, MDD, Editor de Modelado Textual.

## 1 Introducción y Motivación

El procesamiento de eventos complejos (*Complex Event Processing*, CEP) [1] es una tecnología novedosa que permite analizar y correlacionar ingentes volúmenes de datos con el fin de detectar en tiempo real situaciones de interés para un dominio en particular. Para ello se requiere implementar patrones de eventos, especificando las condiciones que deben cumplirse para detectar dichas situaciones, con los lenguajes de procesamiento de eventos (*Event Processing Languages*, EPL). A pesar de que los usuarios suelen tener un vasto conocimiento en el dominio para el que se necesitan definir ciertos patrones de eventos, suelen ser inexpertos tanto en EPL como en el

lenguaje requerido para implementar las acciones a llevar a cabo tras la detección de los patrones.

Un estudio de la empresa ebizQ [2] concluye que un 84% de los encuestados consideran que la definición de estos patrones debe realizarse por expertos en el dominio, que son los que realmente disponen de todo el conocimiento necesario para ello, frente a un 16% que prefieren que sean programadores. El estudio realizado al amparo de la tesis doctoral de Boubeta-Puig [3] también corrobora esta necesidad.

Con objeto de acercar CEP a cualquier usuario eliminando las barreras existentes en cuanto a la programación requerida, Boubeta-Puig et al. han propuesto MEdit4CEP [4], una solución compuesta principalmente por un enfoque dirigido por modelos [5] y un editor gráfico e intuitivo para el modelado de patrones de eventos, así como su transformación automática tanto al código EPL a desplegar en el motor CEP, como al código de las acciones a ejecutar tras la detección de los mismos. Estos patrones se modelan, independientemente de su implementación, a través del lenguaje gráfico de modelado específico de dominio (*Domain-Specific Modeling Language*, DSML) Model4CEP [6].

Dado que algunos usuarios son reticentes a usar herramientas gráficas, en este artículo presentamos un editor de modelado textual que complementa a MEdit4CEP. Este editor supone un avance para aquellos usuarios que prefieren definir los patrones de eventos textualmente – en vez de gráficamente – y sin necesidad de conocer la sintaxis particular de los distintos EPL soportados por los motores CEP, ya que este editor transforma estos modelos textuales de patrones al código que los implementa. Más adelante, se planea sincronizar este editor con MEdit4CEP.

El resto del artículo se estructura de la siguiente forma. En la Sección 2 se propone el editor de modelado textual de patrones de eventos y en la Sección 3 se presentan las conclusiones y el trabajo futuro.

## 2 Editor Textual Propuesto

En esta sección, se describe brevemente cómo se ha llevado a cabo el desarrollo del editor de modelado textual de dominios CEP y patrones de eventos, implementado con Xtext [7] y Xtend [8].

Partiendo de los metamodelos Model4CEP en Ecore y haciendo uso del enfoque de desarrollo dirigido por modelos, la plataforma Xtext ha generado dos gramáticas EBNF (*Domain.xtext* y *EventPattern.xtext*). Entonces, hemos modificado la sintaxis del lenguaje autogenerado por Xtext para tratar de hacerlo lo más sencillo e intuitivo posible de cara al usuario final.

Una vez definidas estas gramáticas, Xtext ha construido automáticamente la infraestructura necesaria para dar soporte a los nuevos lenguajes textuales. Xtext utiliza MWE2 para leer modelos EMF, transformarlos y generar artefactos de código. No obstante, ha sido necesario sobrescribir parte de dicha infraestructura para adecuarla a nuestras necesidades, adaptando las funcionalidades propias de MEdit4CEP para su correcto funcionamiento en este entorno textual. Concretamente, estas funcionalidades son las propias para dominios CEP (creación, auto-detección, validación, etc.), las

de patrones de eventos (creación, validación, transformación a código EPL, etc.), e importación y exportación de los modelos definidos.

Además, Xtext ha generado automáticamente, en la arquitectura creada, mecanismos de autocompletado de contenido. Hemos adaptado esta funcionalidad para otorgar ayuda al usuario que está modelando, de tal forma que si pulsa en la combinación Ctrl + Espacio, se generarán sugerencias de contenido en el punto de modelado donde se encuentra. Por otra parte, se ha sobrescrito la funcionalidad de vista jerárquica del editor textual para que, además de permitir visualizar los elementos definidos en el modelo, se muestren también sus iconos asociados.

Para asegurar la corrección de los modelos, se han definido algunas reglas de validación empleando el lenguaje Xtend. Se ejecutan todas estas reglas cada vez que el modelo sufre algún cambio, avisando al usuario cuando se produzca algún error.

En el video disponible en <http://dx.doi.org/10.17632/zzghkk35gr.1> se muestra paso a paso cómo usar este editor textual en el siguiente caso de estudio. Un termómetro mide cada segundo la temperatura en una planta de energía nuclear y la envía a un sistema de monitorización central; necesitamos definir el patrón de eventos *Monitor*, que notificará la media de las temperaturas cada 10 segundos. Esto se lleva a cabo en dos pasos: (1) creación del dominio CEP denominado *nuclear\_power\_plant*, formado por el tipo de evento *temperatureEvent* con dos propiedades de eventos – *timestamp* y *value* – para registrar en qué instante de tiempo se mide la temperatura en cada momento, y (2) modelado del patrón *Monitor* (véase Figura 1) y su transformación automática a código Esper EPL. Puede observarse en el video que este editor ofrece las características básicas de cualquier buen editor textual, así como todas las funcionalidades adicionales descritas previamente.

### 3 Conclusiones y Trabajo Futuro

En este trabajo hemos propuesto un editor textual para el modelado y la generación de código de dominios CEP y patrones de eventos, basado en MEdit4CEP. Este editor representa una alternativa a MEdit4CEP para que, aquellos usuarios finales que prefieran usar herramientas textuales en vez de gráficas, tengan la posibilidad de definir los patrones sin necesidad de conocer detalles específicos sobre ningún EPL ni sobre ningún lenguaje requerido para implementar las acciones asociadas a estos patrones. Además, este editor valida los patrones modelados y genera el código que los implementa.

Como trabajo futuro, pretendemos integrar este editor textual con MEdit4CEP para ofrecer al usuario final una única herramienta con dos editores de modelado (gráfico y textual) de patrones de eventos que se sincronicen automáticamente.

**Agradecimientos.** Los autores agradecen la financiación del Ministerio de Economía y Competitividad y Fondos FEDER (proyecto TIN2015-65845-C3-3-R) y de la Universidad de Cádiz (proyecto PR2016-032).

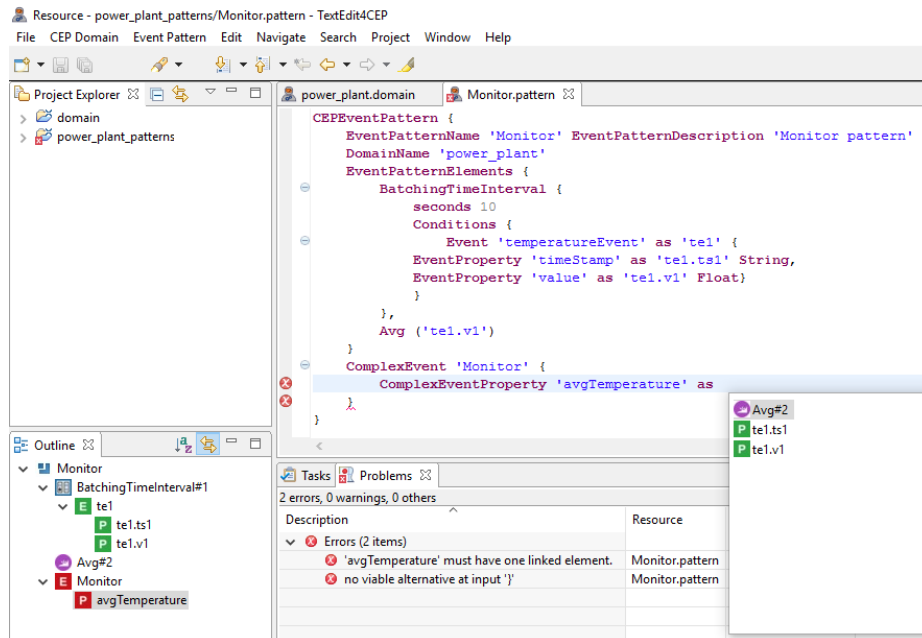


Figura 1. Patrón de eventos *Monitor* modelado con el editor textual propuesto.

## Referencias

- [1] D. Luckham, *Event Processing for Business: Organizing the Real-Time Enterprise*. New Jersey, USA: Wiley, 2012.
- [2] BEA, “Event Processing Market Pulse 2007,” ebizQ, 2007.
- [3] J. Boubeta-Puig, “Desarrollo Dirigido por Modelos de Interfaces Específicas de Dominio para el Procesamiento de Eventos Complejos en Arquitecturas Orientadas a Servicios,” Tesis doctoral. Universidad de Cádiz, Cádiz, España, 2014.
- [4] J. Boubeta-Puig, G. Ortiz, and I. Medina-Bulo, “MEdit4CEP: A model-driven solution for real-time decision making in SOA 2.0,” *Knowledge-Based Systems*, vol. 89, pp. 97–112, Nov. 2015.
- [5] J. Boubeta-Puig, G. Ortiz, and I. Medina-Bulo, “A Model-driven Approach for Facilitating User-friendly Design of Complex Event Patterns,” *Expert Systems with Applications*, vol. 41, no. 2, pp. 445–456, Feb. 2014.
- [6] J. Boubeta-Puig, G. Ortiz, and I. Medina-Bulo, “Model4CEP: Graphical domain-specific modeling languages for CEP domains and event patterns,” *Expert Systems with Applications*, vol. 42, no. 21, pp. 8095–8110, Nov. 2015.
- [7] Eclipse, “Xtext,” [Online] <http://www.eclipse.org/Xtext/>. [Accedido: 18-Feb-2017].
- [8] Eclipse, “Xtend,” [Online] <http://www.eclipse.org/xtend/>. [Accedido: 18-Feb-2017].