

# Procesamiento de Eventos Complejos Aplicado al Ámbito de las Smart Cities

**Dr. Juan Boubeta-Puig**

Departamento de Ingeniería Informática  
Universidad de Cádiz



Universidad Complutense de Madrid  
Madrid, 16 de febrero de 2017





# CONTENIDOS

- Procesamiento de eventos complejos
- Esper EPL
- MEdit4CEP
- Casos de estudio



# CONTENIDOS

- **Procesamiento de eventos complejos**
- Esper EPL
- MEdit4CEP
- Casos de estudio



# PROCESAMIENTO DE EVENTOS COMPLEJOS

## Definición

- Tecnología emergente que permite procesar, analizar y correlacionar grandes cantidades de eventos.
- Para detectar y responder en tiempo real a situaciones críticas o relevantes del negocio.
- Los **patrones de eventos** inferirán nuevos eventos más complejos (“situaciones”) con mayor significado semántico.
- Requisitos software:
  - Motor CEP (*Complex Event Processing*), p.e. Esper.
  - EPL (*Event Processing Language*), p.e. EPL de Esper.





# PROCESAMIENTO DE EVENTOS COMPLEJOS

## Patrón de eventos

### Eventos simples



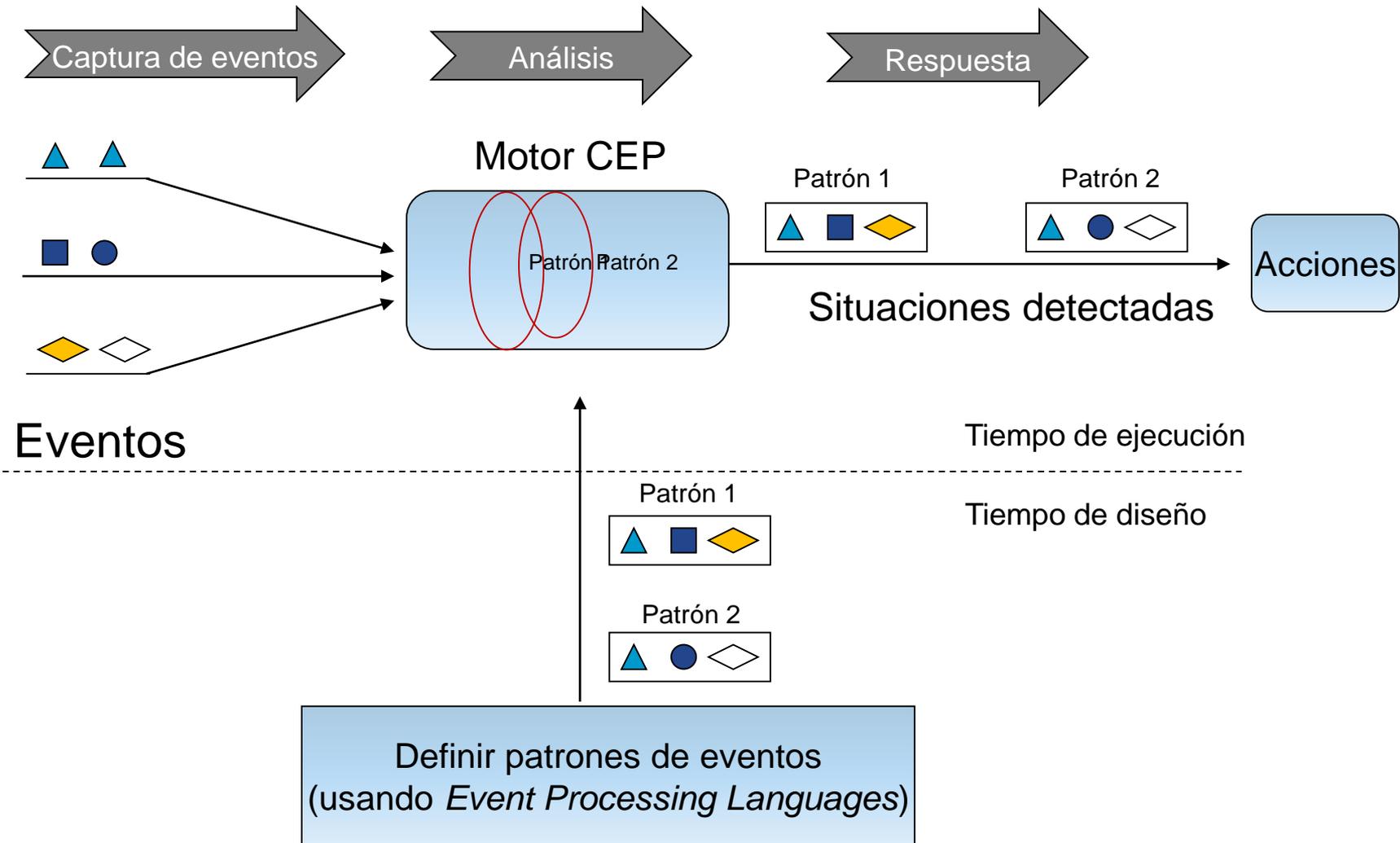
### Evento complejo

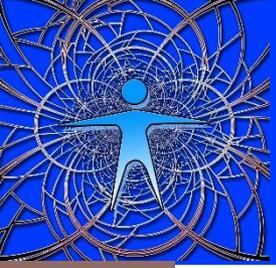




# PROCESAMIENTO DE EVENTOS COMPLEJOS

## Etapas

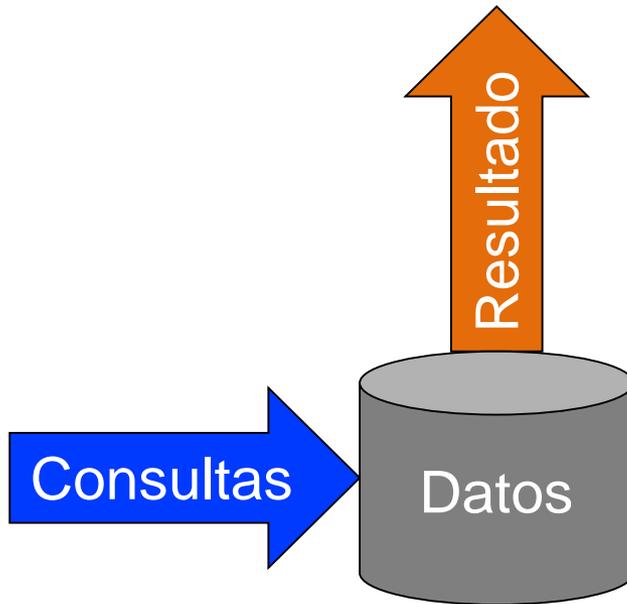




# PROCESAMIENTO DE EVENTOS COMPLEJOS

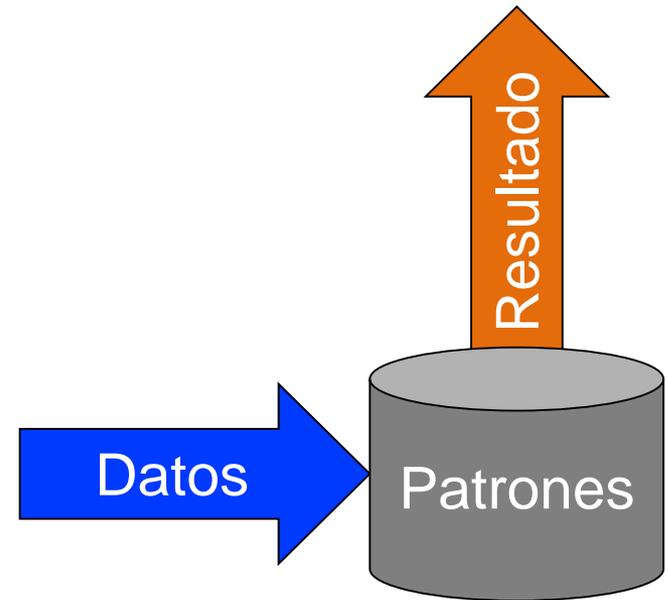
## Procesamiento estático vs dinámico

### Procesamiento estático



La cotización de un valor o una noticia sobre una compañía en un instante determinado

### Procesamiento dinámico: CEP



Si se publica una noticia sobre una compañía y en menos de 30 s la cotización baja un 5%, entonces generar una orden de venta



# PROCESAMIENTO DE EVENTOS COMPLEJOS

## Aplicación práctica



Quiero saber (cuanto antes) qué está pasando y reaccionar de la mejor forma posible

¿Qué ha ocurrido?

*Detect*  
(Detectar)



*Derive*  
(Inferir)



*Decide*  
(Decidir)

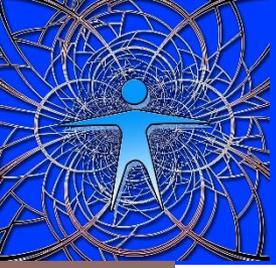


*Do*  
(Actuar)

¿Qué debemos hacer?

*“No va a ser mejor el que más información tenga, sino el que mejor la sepa analizar y más rápidamente actúe”*

(Víctor Ayllón, CEO Novayre Solutions)



# PROCESAMIENTO DE EVENTOS COMPLEJOS

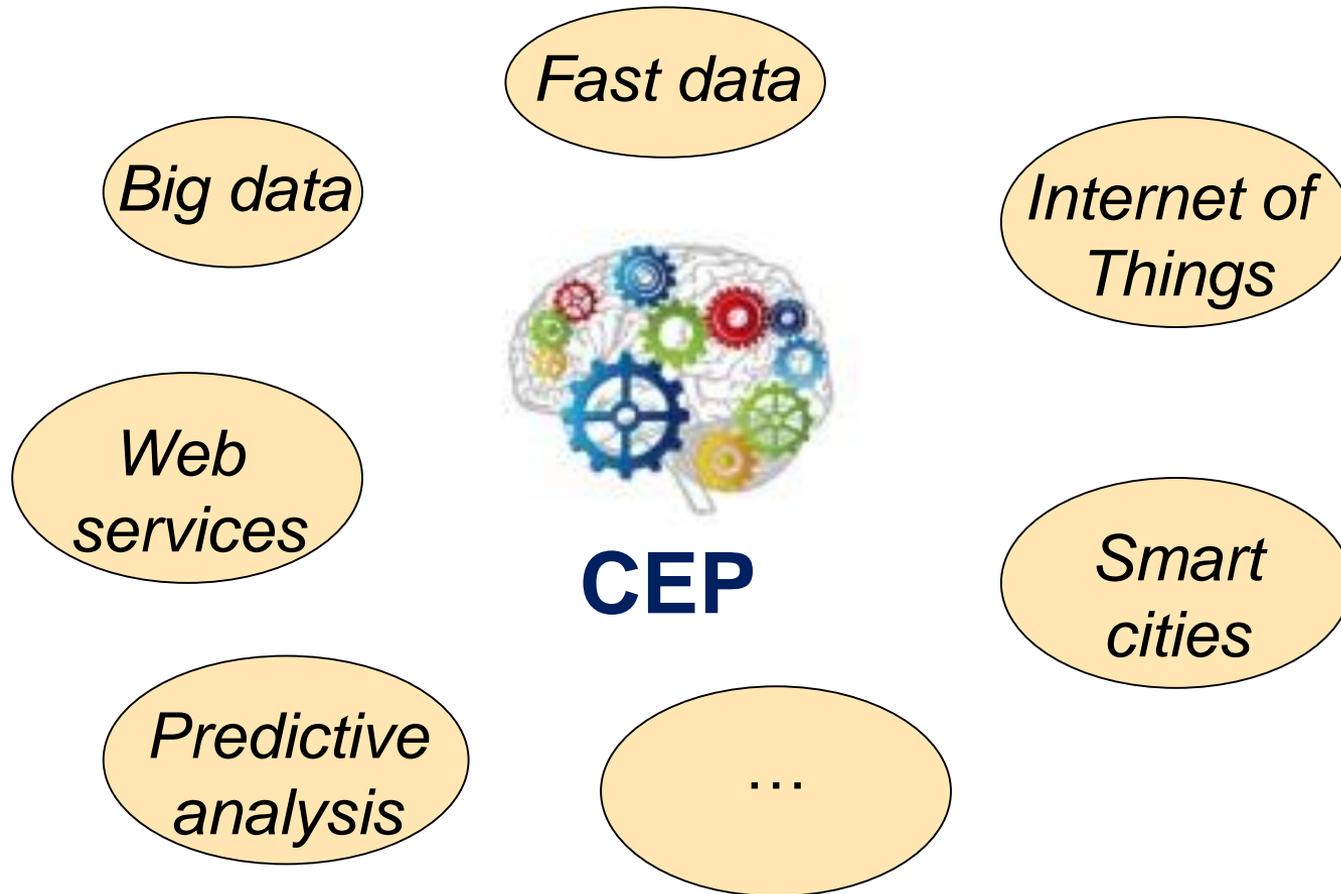
## Ventajas y escenarios de aplicación

- Ventajas:
  - Prevención de sobrecarga de información.
  - Reducción del esfuerzo humano.
  - Respuesta veloz.
  - Mejora de la calidad en las decisiones.
- Escenarios de aplicación:
  - Bolsa: comprar y vender automáticamente.
  - Comercio: detectar compras fraudulentas.
  - IoT: procesar en tiempo real datos de sensores y generar alarmas/acciones sobre actuadores.
  - Seguridad: detectar ataques de seguridad en sistemas informáticos.
  - ...



# PROCESAMIENTO DE EVENTOS COMPLEJOS

## Integración con otros enfoques/tecnologías





# CONTENIDOS

- Procesamiento de eventos complejos
- **Esper EPL**
- MEdit4CEP
- Casos de estudio



# ESPER EPL

## Introducción

- Esper EPL (*Event Processing Language*) es un lenguaje similar a SQL: `SELECT`, `FROM`, `WHERE`, `GROUP BY`...
- Extendido con operadores de patrón, ventanas de datos temporales...
- Tabla (en SQL)  $\Leftrightarrow$  **Flujo de eventos** (en EPL).
- Soporta de forma nativa varios tipos de formato de eventos: objetos Java/.NET, **Java *maps*** y XML.
- Los patrones de eventos implementados en EPL se registran en el motor CEP Esper.
- Al detectarse un patrón se notificará a un *listener*.
- Soporta la creación de **jerarquías de patrones de eventos**.



# ESPER EPL

## Sintaxis (subconjunto)

[*annotations*]

[**insert into** *insert\_into\_def*]

select *select\_list*

**from pattern**

[where *search\_conditions*]

[group by *grouping\_expression\_list*]

[having *grouping\_search\_conditions*]

[order by *order\_by\_expression\_list*]



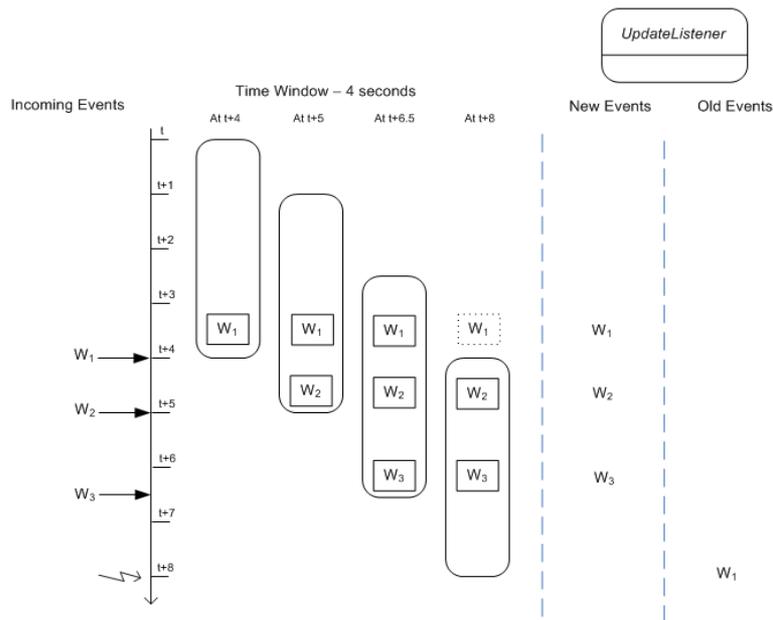
Consideremos la secuencia de eventos:

**A<sub>1</sub> B<sub>1</sub> C<sub>1</sub> B<sub>2</sub> A<sub>2</sub> D<sub>1</sub> A<sub>3</sub> B<sub>3</sub> E<sub>1</sub> A<sub>4</sub> F<sub>1</sub> B<sub>4</sub>**

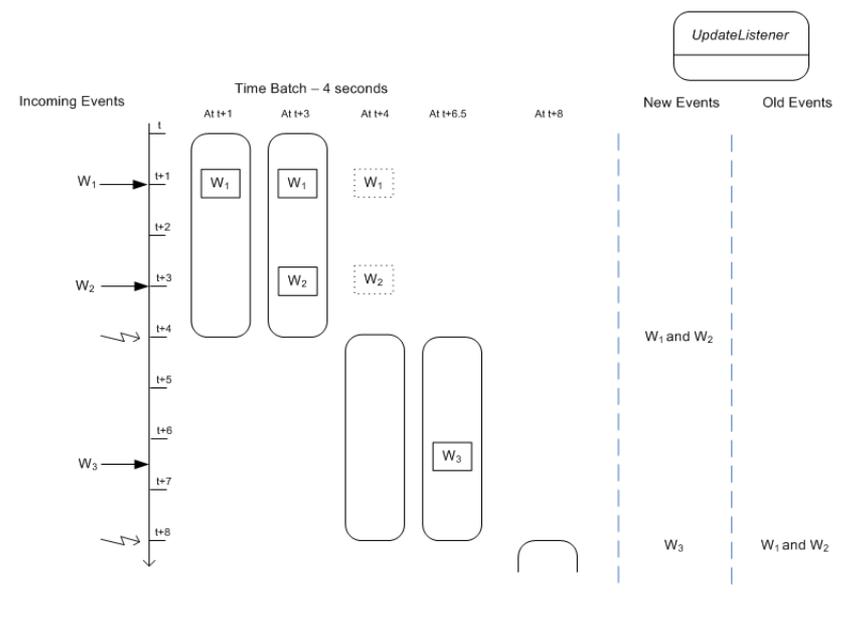
- `every (A -> B)`
  - {A<sub>1</sub>, B<sub>1</sub>}, {A<sub>2</sub>, B<sub>3</sub>}, {A<sub>4</sub>, B<sub>4</sub>}
- `every A -> B`
  - {A<sub>1</sub>, B<sub>1</sub>}, {A<sub>2</sub>, B<sub>3</sub>}, {A<sub>3</sub>, B<sub>3</sub>}, {A<sub>4</sub>, B<sub>4</sub>}
- `A -> every B`
  - {A<sub>1</sub>, B<sub>1</sub>}, {A<sub>1</sub>, B<sub>2</sub>} {A<sub>1</sub>, B<sub>3</sub>} {A<sub>1</sub>, B<sub>4</sub>}
- `every A -> every B`
  - {A<sub>1</sub>, B<sub>1</sub>}, {A<sub>1</sub>, B<sub>2</sub>}, {A<sub>1</sub>, B<sub>3</sub>}, {A<sub>2</sub>, B<sub>3</sub>},  
{A<sub>3</sub>, B<sub>3</sub>}, {A<sub>1</sub>, B<sub>4</sub>}, {A<sub>2</sub>, B<sub>4</sub>}, {A<sub>3</sub>, B<sub>4</sub>},  
{A<sub>4</sub>, B<sub>4</sub>}



### Time Window



### Time Batch Window





# CONTENIDOS

- Procesamiento de eventos complejos
- Esper EPL
- **MEdit4CEP**
- Casos de estudio



# INTRODUCCIÓN

## Motivación (i)

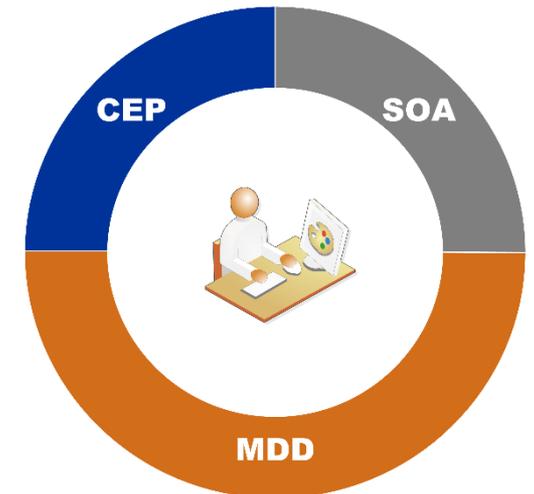
- **Situación actual:**
  - Compañías gestionan grandes cantidades de información.
  - Necesitan conocer el **valor** que tienen los datos para una buena toma de decisiones.
  - Se propone el uso de CEP en sus sistemas de información (SOA).
- **Problema a resolver:**
  - Se requiere a los expertos en el negocio tener conocimientos de programación para:
    - Definir las condiciones de las situaciones a detectar.
    - Definir las acciones a llevar a cabo.
  - Las herramientas actuales no abstraen al usuario de las tecnologías.

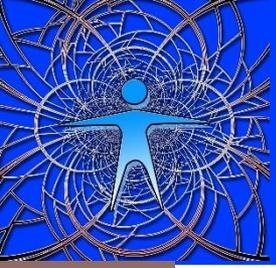


# INTRODUCCIÓN

## Motivación (ii)

- **MEdit4CEP:** Una solución dirigida por modelos para la toma de decisiones en arquitecturas orientadas a servicios y dirigidas por eventos (ED-SOA o SOA 2.0).
  - Integración de CEP con SOA 2.0.
  - Definición de dominios CEP y patrones de eventos mediante editores gráficos e intuitivos.
  - Generación automática de código.
- **Propósito:** poner CEP al alcance de cualquier usuario, repercutiendo beneficiosamente en el proceso de toma de decisiones.



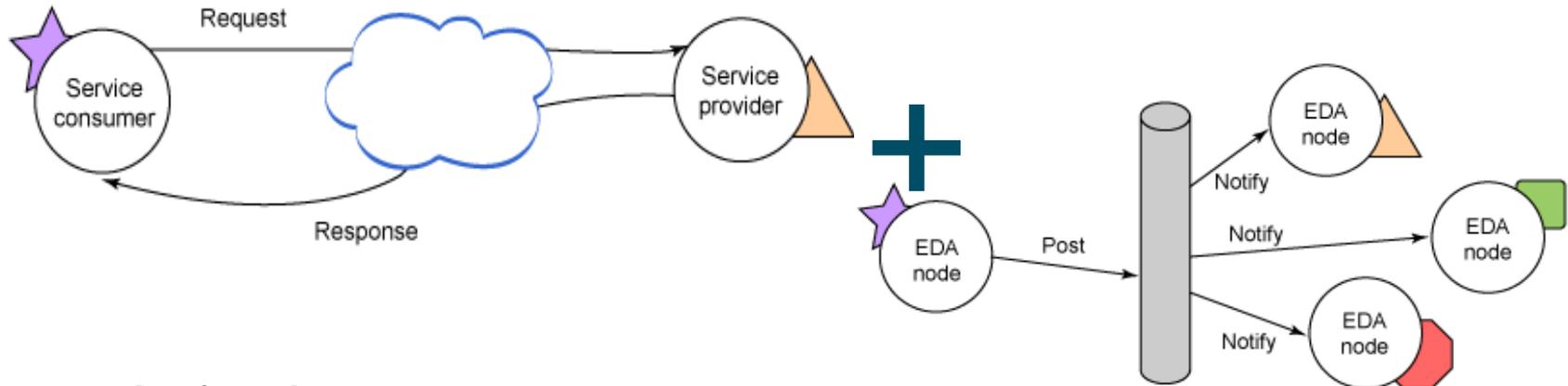


# INTRODUCCIÓN

## SOA 2.0 = SOA + EDA (i)

- Las comunicaciones entre usuarios, servicios y aplicaciones se llevan a cabo a través de eventos.

	SOA	EDA
Interacciones	Débilmente acopladas	Desacopladas
Comunicaciones	Uno a uno	Muchos a muchos (publicación-suscripción)
Reacción frente a	Clientes	Eventos
Tipo comunicación	Síncrono	Asíncrono



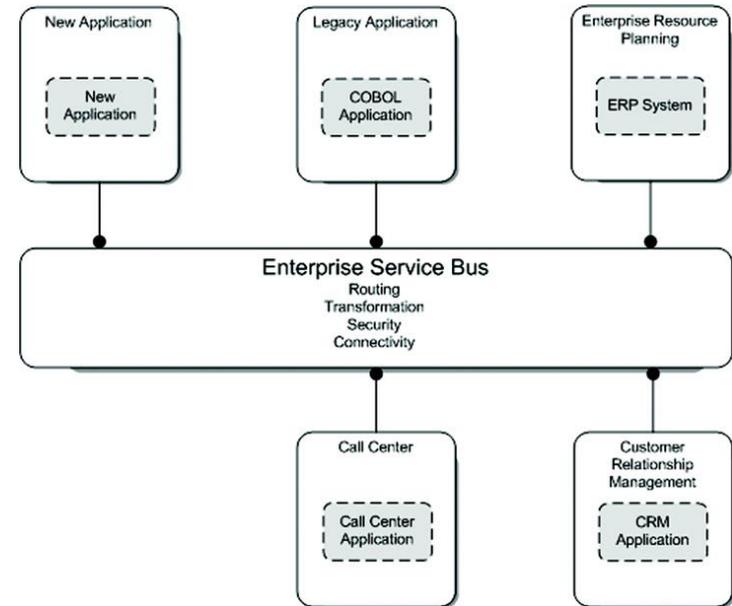
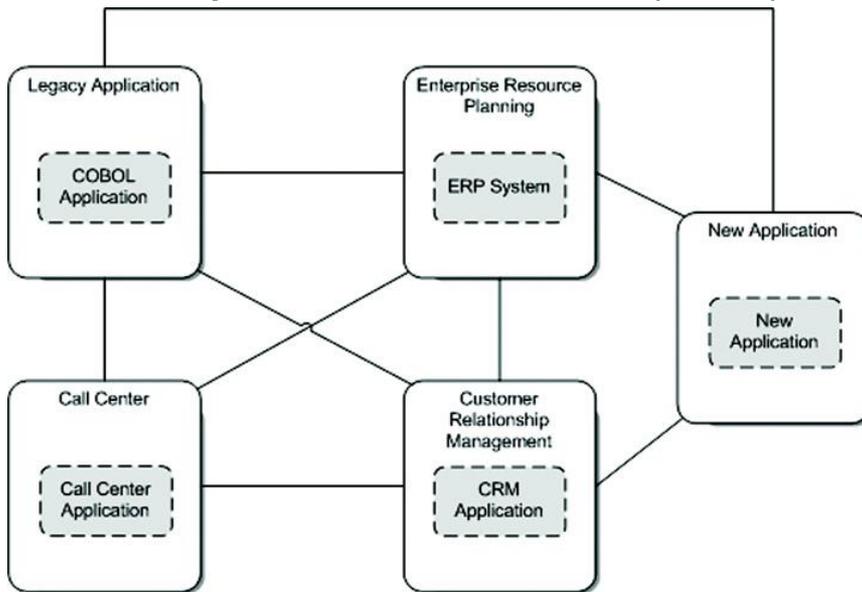
Fuente: [Maréchaux]



# INTRODUCCIÓN

## SOA 2.0 = SOA + EDA (ii)

- Arquitectura tradicional versus arquitectura basada en un *Enterprise Service Bus* (ESB):



Fuente: [Rademakers & Dirksen]

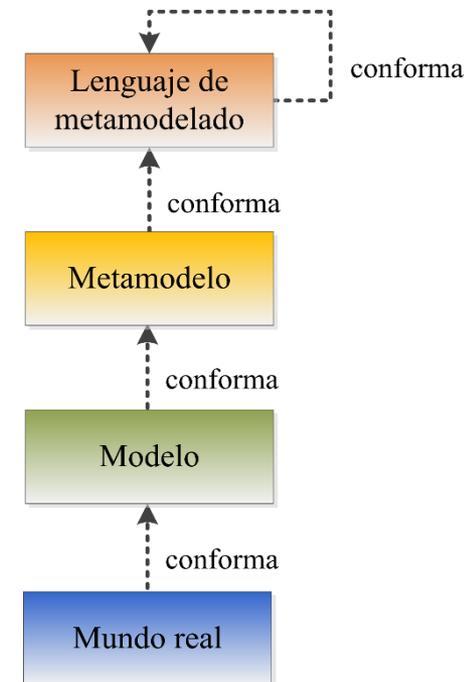
- Ventajas de un ESB:
  - Facilita la integración de SOA y EDA.
  - Ideal para trabajar en entornos heterogéneos.
  - Reduce el coste total de la gestión y el mantenimiento.



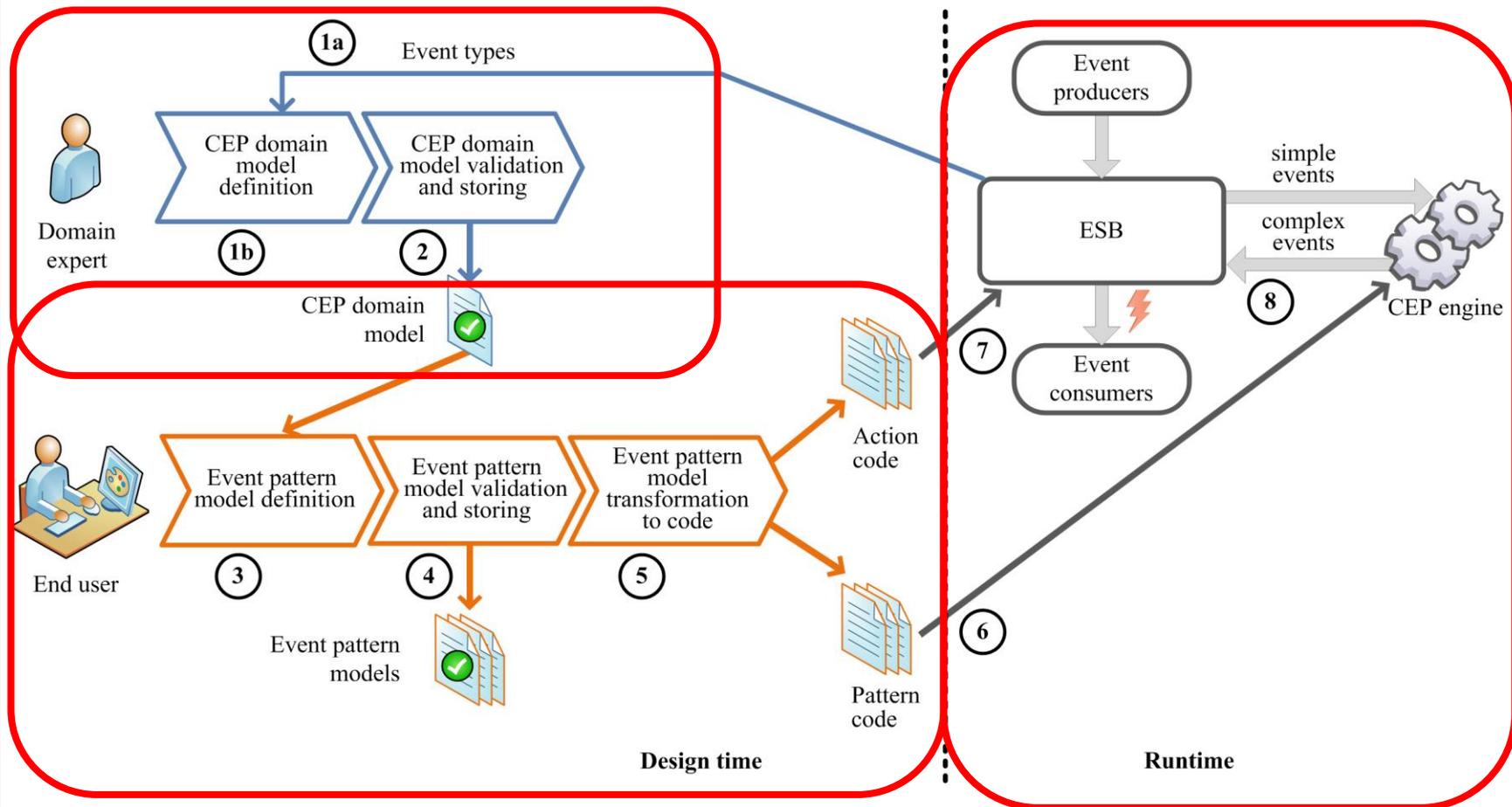
# INTRODUCCIÓN

## Desarrollo de software dirigido por modelos

- **Modelo:** representación simplificada de una determinada realidad para comprenderla mejor.
- **Lenguaje de modelado específico de dominio (DSML)**  
= Sintaxis abstracta (metamodelo y reglas de validación)  
+ sintaxis concreta  
+ transformaciones
- Beneficios de DSML:
  - Mejora la productividad de desarrollo.
  - Mejora la comunicación con expertos en el dominio.
  - Facilita la adaptación ante los cambios.
  - Especifica el qué, no el cómo.



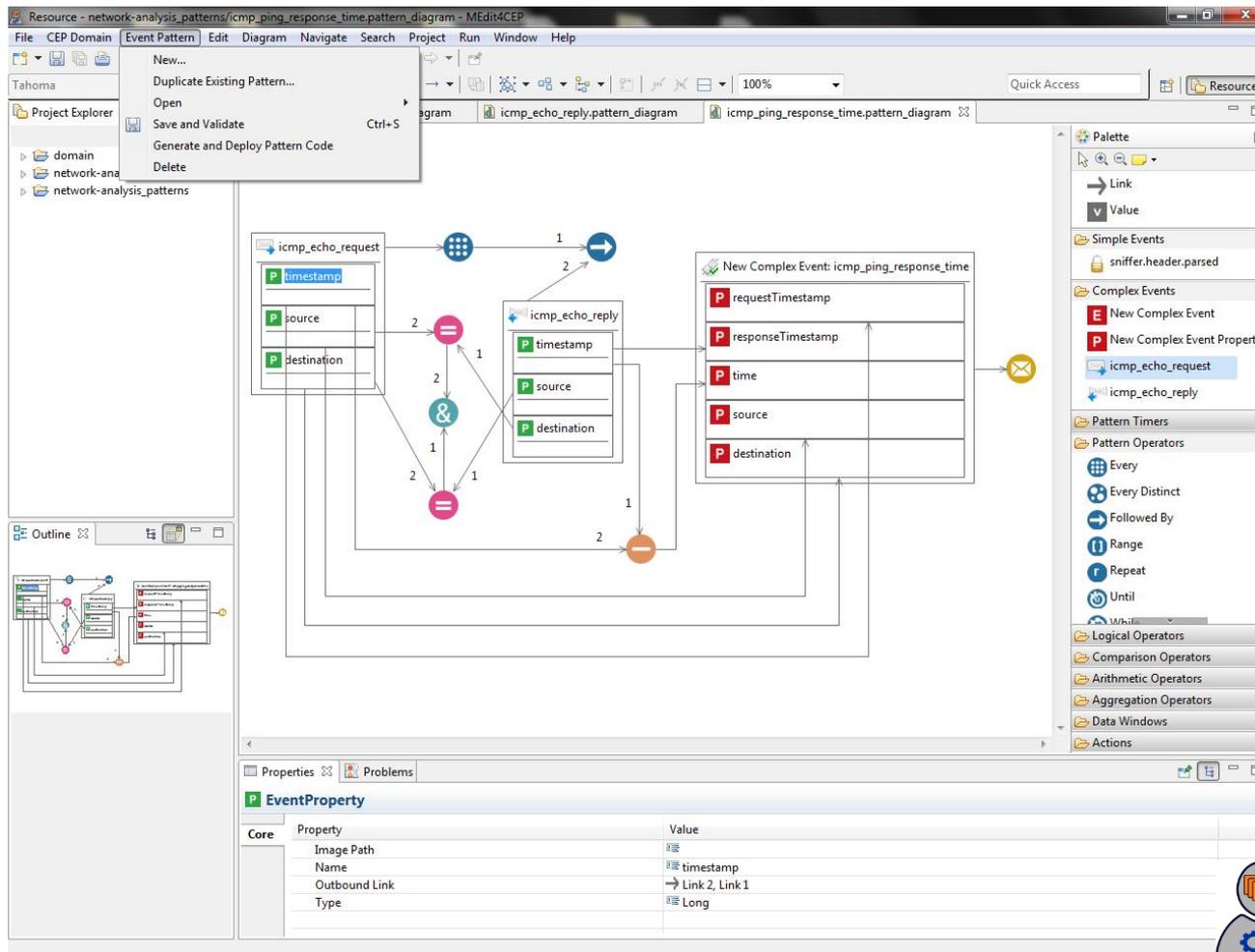
# ENFOQUE DIRIGIDO POR MODELOS PARA CEP EN SOA 2.0



Boubeta-Puig, J., Ortiz, G., & Medina-Bulo, I. (2015). MEdit4CEP: A model-driven solution for real-time decision making in SOA 2.0. Knowledge-Based Systems, 89, 97–112. <http://doi.org/10.1016/j.knosys.2015.06.021>

# EDITOR GRÁFICO DE PATRONES DE EVENTOS

<https://ucase.uca.es/medit4cep/>



Boubeta-Puig, J., Ortiz, G., & Medina-Bulo, I. (2015). MEdit4CEP: A model-driven solution for real-time decision making in SOA 2.0. Knowledge-Based Systems, 89, 97–112. <http://doi.org/10.1016/j.knosys.2015.06.021>



# CONTENIDOS

- Procesamiento de eventos complejos
- Esper EPL
- MEdit4CEP
- **Casos de estudio**

# CASOS DE ESTUDIO

## CEP aplicado a smart cities

### Libelium Smart World

#### Air Pollution

Control of CO<sub>2</sub> emissions of factories, pollution emitted by cars and toxic gases generated in farms.

#### Forest Fire Detection

Monitoring of combustion gases and preemptive fire conditions to define alert zones.

#### Wine Quality Enhancing

Monitoring soil moisture and trunk diameter in vineyards to control the amount of sugar in grapes and grapevine health.

#### Offspring Care

Control of growing conditions of the offspring in animal farms to ensure its survival and health.

#### Sportsmen Care

Vital signs monitoring in high performance centers and fields.

#### Structural Health

Monitoring of vibrations and material conditions in buildings, bridges and historical monuments.

#### Quality of Shipment Conditions

Monitoring of vibrations, strokes, container openings or cold chain maintenance for insurance purposes.

#### Smartphones Detection

Detect iPhone and Android devices and in general any device which works with Wifi or Bluetooth interfaces.

#### Perimeter Access Control

Access control to restricted areas and detection of people in non-authorized areas.

#### Radiation Levels

Distributed measurement of radiation levels in nuclear power stations surroundings to generate leakage alerts.

#### Electromagnetic Levels

Measurement of the energy radiated by cell stations and and WiFi routers.

#### Traffic Congestion

Monitoring of vehicles and pedestrian affluence to optimize driving and walking routes.

#### Smart Roads

Warning messages and diversions according to climate conditions and unexpected events like accidents or traffic jams.

#### Smart Lighting

Intelligent and weather adaptive lighting in street lights.

#### Intelligent Shopping

Getting advices in the point of sale according to customer habits, preferences, presence of allergic components for them or expiring dates.

#### Noise Urban Maps

Sound monitoring in bar areas and centric zones in real time.

#### Water Leakages

Detection of liquid presence outside tanks and pressure variations along pipes.

#### Vehicle Auto-diagnosis

Information collection from CanBus to send real time alarms to emergencies or provide advice to drivers.

#### Item Location

Search of individual items in big surfaces like warehouses or harbours.

#### Waste Management

Detection of rubbish levels in containers to optimize the trash collection routes.

#### Smart Parking

Monitoring of parking spaces availability in the city.

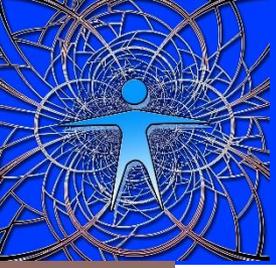
#### Golf Courses

Selective irrigation in dry zones to reduce the water resources required in the green.

#### Water Quality

Study of water suitability in rivers and the sea for fauna and eligibility for drinkable use.

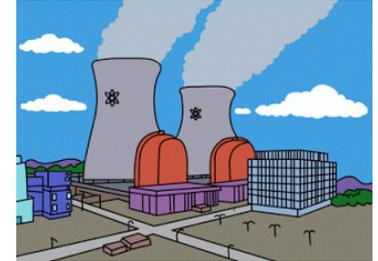




# PLANTA DE ENERGÍA NUCLEAR

## Descripción

- Termómetro que toma una lectura de temperatura cada segundo y la envía a un sistema de monitorización central.
- Necesitamos recibir notificaciones cuando se detecten 3 tipos de **eventos complejos**:
  - **Monitor**: media de la temperatura cada 10 segundos.
  - **Warning**: 2 lecturas de temperatura consecutivas superior a 400 °C.
  - **Critical**: 4 lecturas de temperatura consecutivas, donde la primera es mayor de 100 °C y cada una de las siguientes es mayor a la anterior. Además, la última es 1,5 veces superior a la primera.



Fuente: <http://www.adrianmilne.com/complex-event-processing-made-easy/>



# PLANTA DE ENERGÍA NUCLEAR

## Modelado del dominio CEP

The screenshot shows the MEdit4CEP software interface. The main window displays a diagram of a 'TemperatureEvent' object with two properties: 'timestamp' and 'value'. The 'Properties' panel at the bottom shows the 'EventProperty' table with the following data:

Core	Property	Value
Appearance	Image Path	
	Name	value
	Type	Float



# PLANTA DE ENERGÍA NUCLEAR

## Modelado del patrón *Monitor*

Resource - nuclear\_power\_plant\_patterns/Monitor.pattern\_diagram - MEdit4CEP

File CEP Domain Event Pattern Edit Diagram Navigate Search Project Run Window Help

Segoe UI 9 B I A Quick Access Resource

Project Explorer

- domain
- nuclear\_pow

Batching Time Interval: 10s

TemperatureEvent

- timestamp
- value

New Complex Event: Monitor

- avg\_temp

Palette

- Link
- Value
- Simple Events
- Complex Events
  - New Complex Event
  - New Complex Event Property
- Pattern Timers
- Pattern Operators
- Logical Operators
- Comparison Operators
- Arithmetic Operators
- Aggregation Operators
- Data Windows
- Actions

Outline

Tasks Properties

Link

Core	Property	Value
Appearance	Operand	Avg
	Operator	Complex Event Property avg_temp
	Order	1

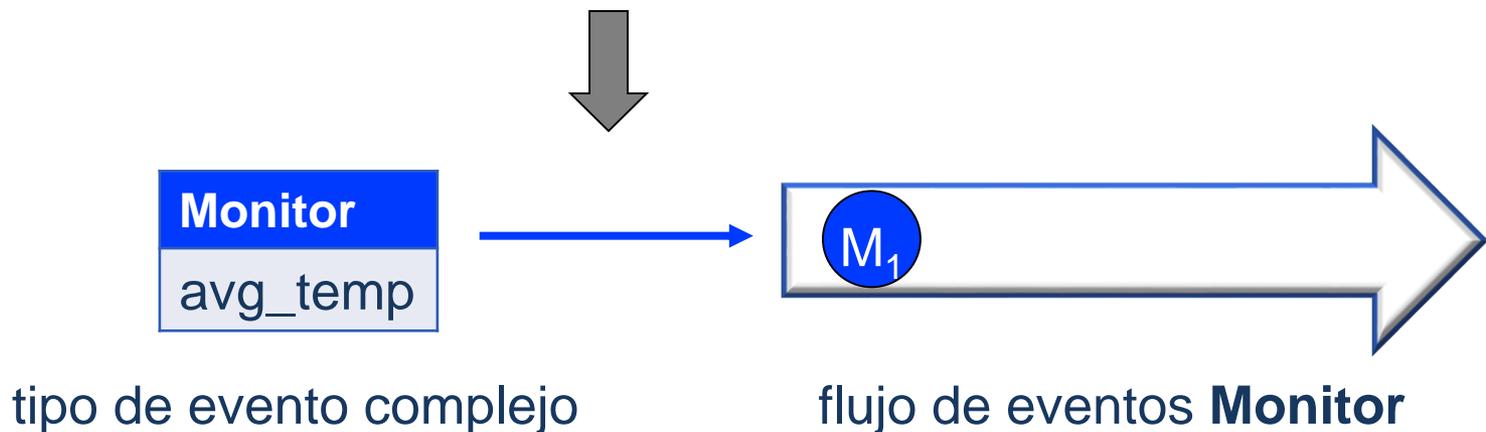


# PLANTA DE ENERGÍA NUCLEAR

## Transformación del patrón *Monitor* a código

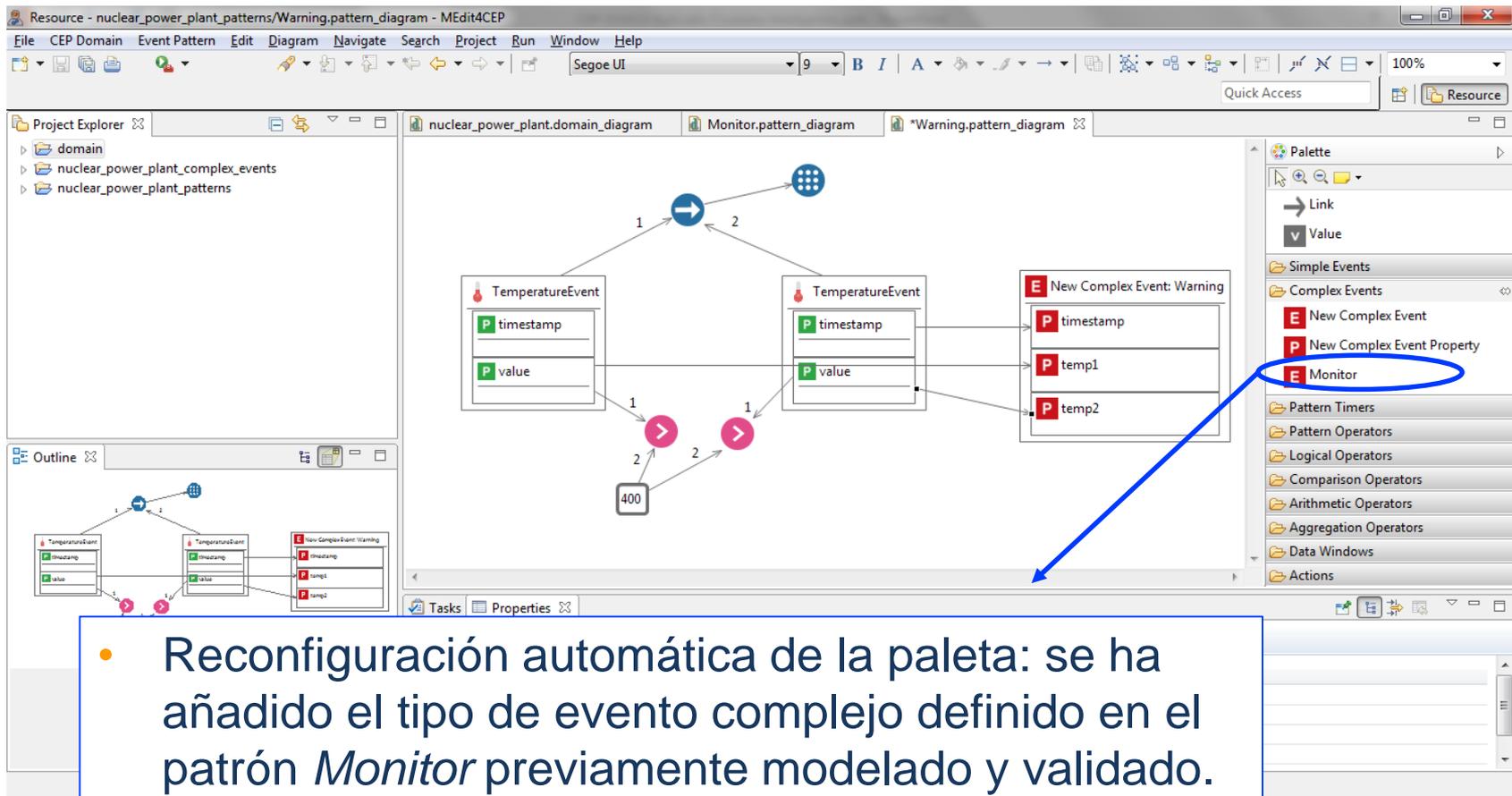
- Código Esper EPL generado automáticamente:

```
@Name ("Monitor")  
insert into Monitor  
select avg(a1.value) as avg_temp  
from TemperatureEvent.win:time_batch(  
    10 seconds) a1
```



# PLANTA DE ENERGÍA NUCLEAR

## Modelado del patrón *Warning*



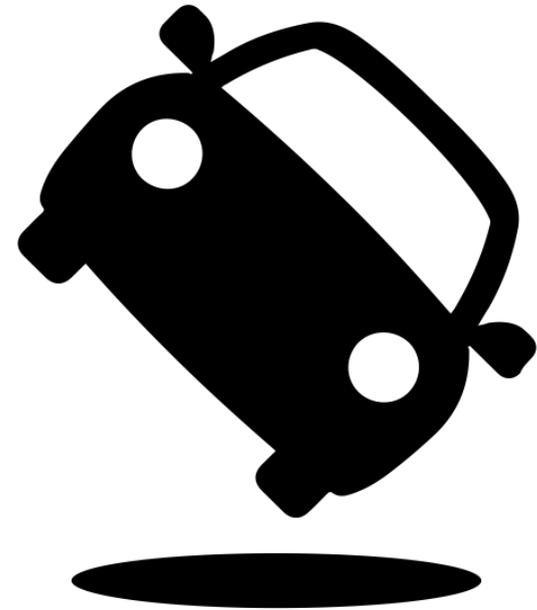
- Reconfiguración automática de la paleta: se ha añadido el tipo de evento complejo definido en el patrón *Monitor* previamente modelado y validado.
- Esto permite el modelado de una jerarquía de patrones de eventos.



# TRÁFICO DE CARRETERAS

## Descripción

- Sensores en un coche que miden:
  - Presión de las ruedas.
  - Velocidad del coche.
  - Asiento ocupado o libre.



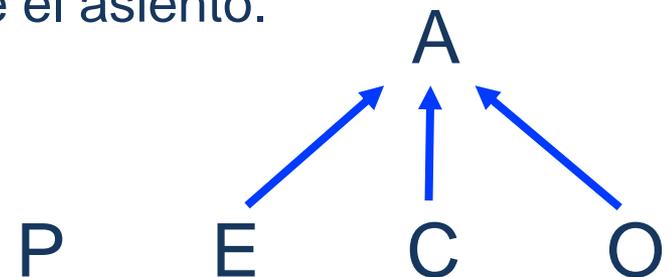
Created by Gregor Črešnar  
from Noun Project



# TRÁFICO DE CARRETERAS

## Patrones de eventos

- **PerdidaPresionRueda (P)**: la presión de una rueda pasa de 3,1 BAR a  $\leq 2,8$  BAR en menos de 15 minutos.
- **ExplosionRueda (E)**: la presión de una rueda pasa de 3,1 BAR a  $\leq 1,2$  BAR en menos de 5 segundos.
- **Colision (C)**: la velocidad del coche superior a 50 km/h pasa a 0 km/h en menos de 3 segundos.
- **OcupanteAbandonaAsiento (O)**: el asiento ocupado pasa a estar libre.
- **AccidenteOcupanteDespedido (A)**: se explota una rueda y, en los siguientes 5 segundos, se produce una colisión provocando que el ocupante abandone el asiento.

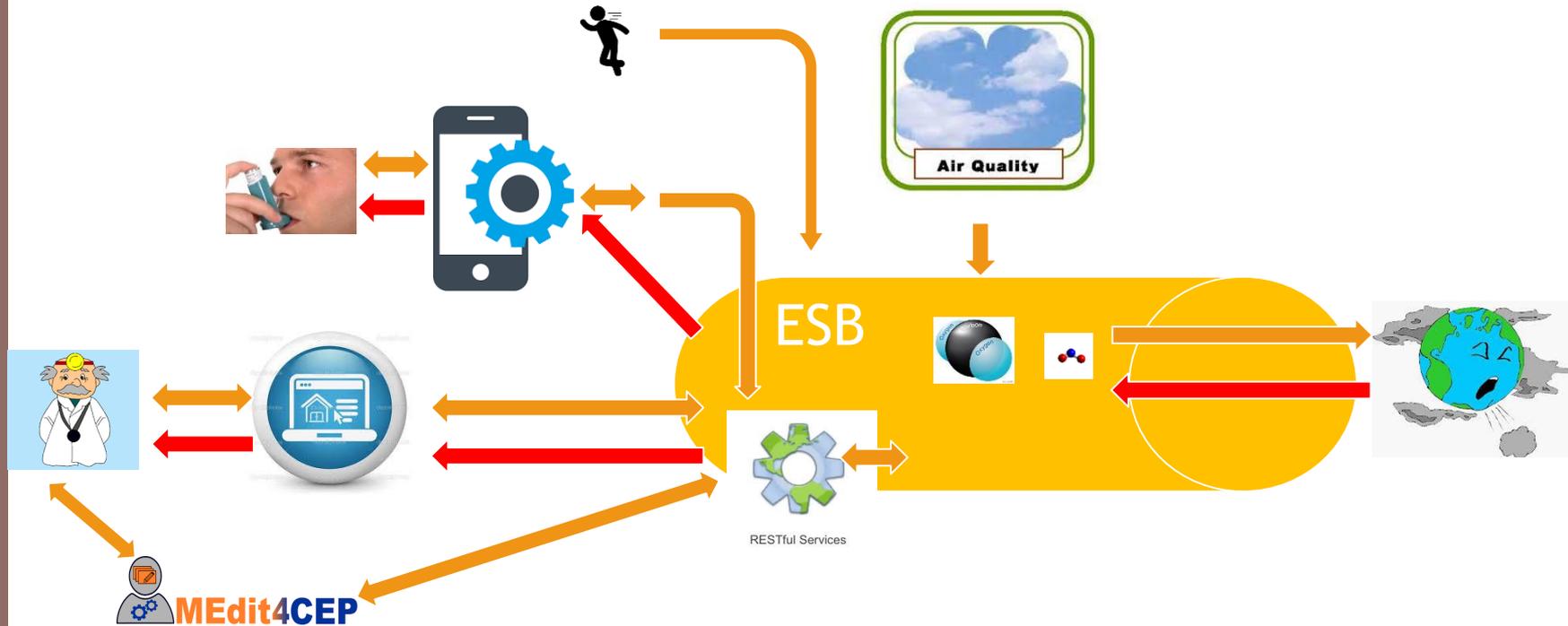




# CALIDAD DEL AIRE

## Descripción

Caso de estudio en colaboración UCM & UCLM & UCA  
(proyecto TIN2015-65845-C3 coordinado por Prof. Manuel Núñez).

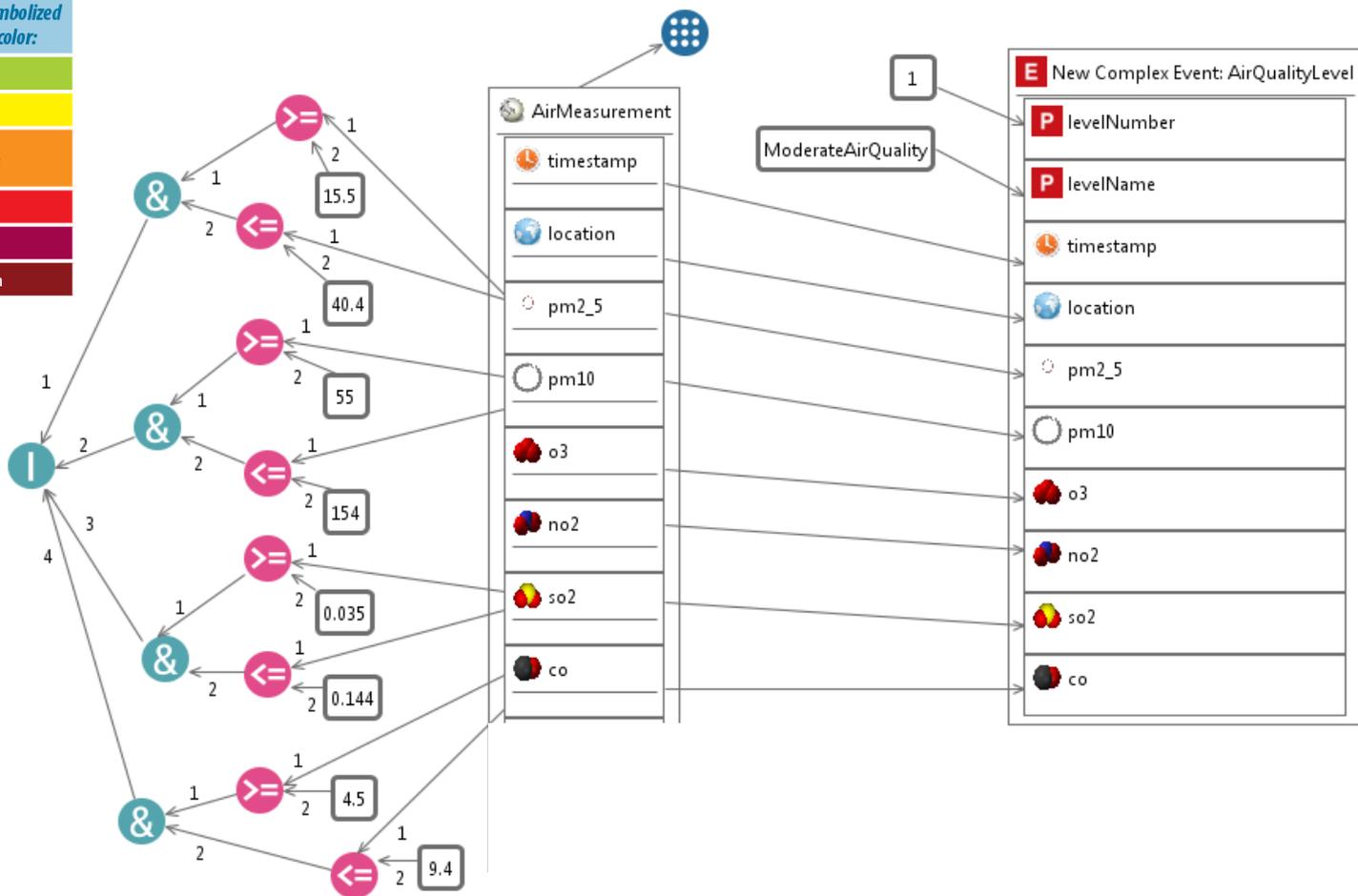




# CALIDAD DEL AIRE

## Modelado del patrón (simplificado) *ModerateAirQuality*

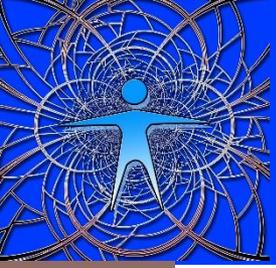
Air Quality Index (AQI) Values	Levels of Health Concern	Colors
When the AQI is in this range:	...air quality conditions are:	...as symbolized by this color:
0 to 50	Good	Green
51 to 100	Moderate	Yellow
101 to 150	Unhealthy for Sensitive Groups	Orange
151 to 200	Unhealthy	Red
201 to 300	Very Unhealthy	Purple
301 to 500	Hazardous	Maroon





# OTROS





# REFERENCIAS BIBLIOGRÁFICAS

## Tesis doctoral:

- Boubeta-Puig, J. *Desarrollo Dirigido por Modelos de Interfaces Específicas de Dominio para el Procesamiento de Eventos Complejos en Arquitecturas Orientadas a Servicios*. Universidad de Cádiz, Cádiz, España, 2014. <http://rodin.uca.es:80/xmlui/handle/10498/17554>

## CEP & SOA 2.0:

- Boubeta-Puig; Ortiz, G; Medina-Bulo, I. *MEdit4CEP: A Model-driven Solution for Real-time Decision Making in SOA 2.0*. Knowledge-Based Systems, vol. 89, pp. 97–112, Nov. 2015.
- Boubeta-Puig; Ortiz, G; Medina-Bulo, I. *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.
- Boubeta-Puig; Ortiz, G; Medina-Bulo, I. *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.



# REFERENCIAS BIBLIOGRÁFICAS

## CEP:

- Chandy, K.M.; Schulte, W.R. *Event Processing: Designing IT Systems for Agile Companies*. McGraw-Hill, 2010.
- Etzion, O.; Niblett, P. *Event Processing in Action*. Manning, 2010.
- Luckham, D. *Event Processing for Business: Organizing the Real-Time Enterprise*. Wiley, 2012.
- Luckham, D. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley, 2001.

## SOA:

- Papazoglou, M.P. *Web Services & SOA: Principles and Technology*. Pearson – Prentice Hall, 2012.

## EDA:

- Taylor, H.; Yochem, Y.; Phillips, L.; Martinez, F. *Event-Driven Architecture: How SOA Enables the Real-time Enterprise*. Addison-Wesley, 2008.



# REFERENCIAS BIBLIOGRÁFICAS

## Esper:

- EsperTech Inc. *Documentation: Esper & EsperIO*, 2017.  
<http://www.espertech.com/esper/documentation.php>
- EsperTech Inc. *Documentation: Chapter 7. EPL Reference: Patterns*, 2017.  
[http://www.espertech.com/esper/release-5.3.0/esper-reference/html/event\\_patterns.html](http://www.espertech.com/esper/release-5.3.0/esper-reference/html/event_patterns.html)
- EsperTech Inc. *Esper EPL Online*, 2017.  
<http://esper-epl-tryout.appspot.com/epltryout/index.html>

## MEdit4CEP:

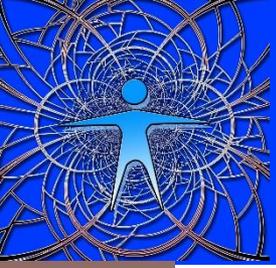
- J. Boubeta-Puig, 2017. <https://ucase.uca.es/medit4cep/>

## Mule:

- MuleSoft Inc. *Anypoint Studio*, 2017.  
<https://www.mulesoft.com/platform/studio>

## ThingSpeak:

- *ThingSpeak*, 2017. <https://thingspeak.com/>
- *ThingSpeak Documentation*, 2017.  
<https://es.mathworks.com/help/thingspeak/>



¡Muchas gracias por su atención!

<https://ucase.uca.es/juan-boubeta-puig>

juan.boubeta@uca.es

