

Sistemas Activos

Lorena CHAVARRÍA-BÁEZ

Departamento de Ingeniería en Sistemas Computacionales, Instituto Politécnico Nacional – Escuela Superior de Cómputo
Ciudad de México, C. P. 07738, México
lchavarria@ipn.mx

Rosaura PALMA-OROZCO

Departamento de Ciencias e Ingeniería de la Computación, Instituto Politécnico Nacional – Escuela Superior de Cómputo
Ciudad de México, C. P. 07738, México
rpalma@ipn.mx

Elena F. RUIZ-LEDESMA

Departamento de Formación Básica, Instituto Politécnico Nacional – Escuela Superior de Cómputo
Ciudad de México, C. P. 07738, México
efruiz@ipn.mx

RESUMEN

Los sistemas activos permiten reaccionar de manera automática ante ciertos eventos relevantes para una aplicación. Este comportamiento es posible gracias a las reglas activas, las cuales especifican el conocimiento que se tiene acerca de un dominio. En este trabajo se abordan los sistemas activos y varios aspectos importantes que intervienen en su desarrollo.

Palabras Claves: Sistemas activos, reglas activas, interacción de reglas.

1. INTRODUCCIÓN

Los *sistemas activos* reaccionan ante eventos, que pueden suceder dentro o fuera del sistema, ejecutando acciones automáticamente [1]. Por ejemplo, considérese una aplicación que está supervisando el cambio en la intensidad de la luz natural para entonces ajustar la intensidad de la luz dentro de una habitación. Este procedimiento necesita ser ejecutado automáticamente para brindar confort al usuario y/o ahorrar energía eléctrica. Aplicaciones como la descrita pueden desarrollarse expresando el conocimiento que se tiene de la aplicación en forma de reglas, es decir, la situación previa puede escribirse como “cuando cambie la intensidad de la luz natural, si la intensidad de la luz está por debajo de un cierto umbral, entonces incrementar la intensidad de la luz de la habitación”, e incorporando tales reglas al desarrollo del sistema completo. Una regla de la forma anterior (*cuando* evento - *si* condición - *entonces* acción) se conoce como *regla activa* o *regla Evento - Condición - Acción* (ECA) y es la parte medular de los sistemas activos.

A primera vista, parece sencillo lograr que un sistema tenga un comportamiento reactivo a través de la introducción de reglas activas. Sin embargo, esta tarea se complica si cuando se consideran aspectos como los siguientes: primero, qué partes de la aplicación pueden desarrollarse mediante reglas activas y cuál es el costo que se debe pagar por el empleo de las mismas; segundo, la expresividad del lenguaje de eventos, lo cual permite definir el número y tipo de eventos que pueden ser supervisados, y, tercero, la evaluación que se hará de la

condición y la acción con respecto al evento, es decir, cuánto tiempo después de que el evento ha sucedido se debe evaluar la condición y ejecutar la acción.

En este trabajo se presenta una descripción de los sistemas activos. El documento está organizado de la siguiente forma: la Sección 2 introduce las reglas activas, las cuales son una forma de representación de conocimiento, y la forma en la que estas se ejecutan. La Sección 3 describe la interacción de las reglas. La Sección 4 presenta algunas aplicaciones. Finalmente, se muestran las conclusiones y las referencias.

2. REGLAS ACTIVAS

Generalmente, una *regla activa*, también conocida como *regla ECA* (*Evento – Condición – Acción*), tiene la forma [1], [2]:

ON evento
IF condición
THEN acción

Esta sintaxis le permite responder de forma automática, mediante la ejecución de acciones predefinidas, a eventos relevantes para el sistema siempre que se cumplan algunas condiciones. Una base de reglas, R, está formada por varias reglas activas.

Cada elemento de una regla ECA posee características que enriquecen su poder de expresión de conocimiento. A continuación se detalla cada elemento que conforma a una regla ECA.

De acuerdo a la Real Academia Española, un evento es una “cosa que sucede” [8]. Puede originarse por alguna operación sobre una base de datos (BD), por ejemplo, insertar una tupla o actualizar un atributo, por la ejecución de operaciones definidas por el usuario, por comandos de transacciones, por ejemplo, el comando abort, por una situación fuera de la BD, por ejemplo, la temperatura subió por encima de 30 grados. Un evento puede ser de dos tipos: primitivo o compuesto [2]. En el primer caso, el evento no puede dividirse en partes más pequeñas. En el

segundo, el evento se forma por la combinación de eventos primitivos, o compuestos, mediante un conjunto de operadores que constituyen el *álgebra de eventos*.

La condición de una regla evalúa el contexto en el cual el evento tomó lugar [1]. El contexto indica el marco en el cual será evaluada la condición.

Las acciones son las tareas que se pueden ejecutar como respuesta a la ocurrencia del evento. Estas pueden ser [4]: actualizar la estructura de una BD, realizar la invocación, interna o externa, de algún comportamiento específico, informar al usuario o al administrador del sistema de alguna situación anómala, interrumpir una transacción.

Modelo de ejecución

El modelo de ejecución indica la forma en la que un conjunto de reglas se trata en tiempo de ejecución. En general, se consideran las siguientes fases en la evaluación de una regla [1]:

- Señalización, se refiere a la ocurrencia de un evento debido a una fuente de evento.
- Disparo, toma los eventos producidos hasta el momento y activa las reglas correspondientes.
- Evaluación, evalúa la condición de la regla que ha sido activada. Si existen varias instancias de reglas y de todas ellas se satisface su condición entonces se tiene un conjunto de reglas en conflicto ya que todas ellas pueden ejecutar su acción al mismo tiempo.
- Calendarización, indica la forma de resolver los conflictos en el conjunto anterior.
- Ejecución, realiza las acciones de la instancia de regla escogida. Durante la ejecución de la acción otros eventos pueden ser señalados lo que puede producir el disparo de reglas en cascada.

La figura 1 muestra la relación entre cada una de estas etapas.

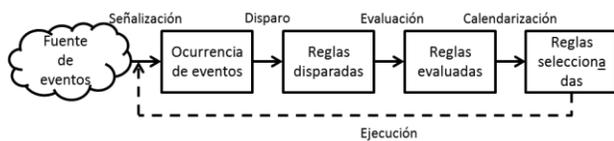


Figura 1. Etapas principales en la ejecución de una regla

Ejemplo

A continuación se muestran algunas reglas activas tomadas de la referencia [3]. Estas reglas son parte de un sistema de BD activa, el cual consiste una una BD tradicional y una base de reglas. Las reglas están definidas sobre el esquema de relación *account(num, name, balance, rate)* el cual almacena información de las cuentas bancarias de los clientes. Es importante notar que el atributo *rate* se encuentra en el intervalo [0, 10]. Las reglas verifican automáticamente el cumplimiento de algunas políticas establecidas por el banco.

Política 1: Cuando se registra una nueva cuenta, si su balance es menor que \$500 y su tasa de interés es mayor a 0%, entonces la tasa de interés de esa cuenta se fija en 0%.

Política 2: Cuando se modifica la tasa de interés de una cuenta, si actualmente esta tiene un balance menor a \$500 y su tasa de

interés es mayor a 0%, entonces la tasa de interés de la cuenta se actualiza a 0%.

Política 3: Cuando se modifica el valor del balance de una cuenta, si la tasa de interés de esa cuenta es mayor a 1% y menor a 3%, entonces la tasa de interés de la cuenta se actualiza a 2%.

Política 4: Cuando se modifica el valor del balance de una cuenta, si la tasa de interés de esa cuenta es mayor a 1% y menor a 3%, entonces la cuenta es eliminada de la relación *account*.

Las políticas anteriores se pueden representar como reglas activas de la forma siguiente:

Regla 1

ON insert account
IF insert.balance < 500 and insert.rate > 0.0
THEN update account set rate = 0.0
 where balance < 500 and rate > 0.0

Regla 2

ON update account.rate
IF update.balance < 500 and update.rate > 0.0
THEN update account set rate = 0.0
 where balance < 500 and rate > 0.0

Regla 3

ON update account.balance
IF update.rate > 1 and update.rate < 3.0
THEN update account set rate = 2.0
 where rate > 1 and rate < 3.0

Regla 4

ON update account.balance
IF update.rate > 1 and update.rate < 3.0
THEN delete from account
 where rate > 1 and rate < 3.0

Considérese la relación *account* con la información mostrada en la tabla 1. Supóngase que se ejecuta la operación “update account set balance = 400 where update.rate > 1 and update.rate < 3”. Este procedimiento asigna el valor 400 al atributo *balance* en la relación *account* de todas las tuplas cuyo valor en el atributo *rate* sea mayor a 1 y menor a 3. De forma que las tuplas con *num* 1, 2 y 3 ahora tienen el valor 400 en su atributo *balance*.

Tabla 1. Contenido de la relación *account*

Account			
num	name	Balance	rate
1	Juan Pérez	1000	1.5
2	Luis Martínez	1800	2.5
3	María Sánchez	2000	2.7
4	Ivonne López	10000	3.0
5	Mario Ibarra	300	0.0
6	Karla Hernández	100	0.0

La operación anterior genera la detección el evento *update account.balance* (señalización), el cual dispara la Regla 3 y la Regla 4 (disparo). Posteriormente, se evalúa la condición de ambas reglas (evaluación) para determinar el conjunto de

acciones a ejecutar. El resultado de la evaluación es verdadero debido a que existen tuplas que poseen un valor en el atributo *rate* mayor a 1 y menor a 3, por lo tanto, se debe llevar a cabo la acción de ambas reglas. Por simplicidad, considérese el orden en que aparecen las reglas en la lista para ejecutar sus acciones (calendarización). De esta manera, primero se efectúa la acción de la Regla 3, es decir, a todas las tuplas de la relación *account* que tengan un valor entre 1 y 3 en el atributo *rate*, se les asigna el valor 2 en ese atributo. El nuevo estado de la relación se muestra en la tabla 2.

Tabla 2. Contenido de la relación *account* después de ejecutar la acción de la Regla 3

Account			
<i>num</i>	<i>name</i>	<i>balance</i>	<i>rate</i>
1	Juan Pérez	400	2.0
2	Luis Martínez	400	2.0
3	María Sánchez	400	2.0
4	Ivonne López	10000	3.0
5	Mario Ibarra	300	0.0
6	Karla Hernández	100	0.0

La acción anterior trae como consecuencia la ocurrencia y detección del evento *update account.rate* el cual dispara la Regla 2. La evaluación de su condición es verdadera ya que existen tuplas cuyo valor de los atributos *balance* y *rate* es menor a 500 y mayor a 0, respectivamente. Por lo tanto, se ejecuta la acción de la Regla 2 modificando el contenido de la relación *account* como se muestra en la tabla 3.

Tabla 3. Contenido de la relación *account* después de ejecutar la acción de la Regla 2

Account			
<i>num</i>	<i>name</i>	<i>balance</i>	<i>rate</i>
1	Juan Pérez	400	0.0
2	Luis Martínez	400	0.0
3	María Sánchez	400	0.0
4	Ivonne López	10000	3.0
5	Mario Ibarra	300	0.0
6	Karla Hernández	100	0.0

La Regla 2 se dispara nuevamente ya que su acción coincide con su evento. Sin embargo, en esta ocasión la evaluación de su condición es falsa ya que no existen tuplas en la relación *account* que tengan un valor mayor a 0 en su atributo *rate*. De esta manera, termina el procesamiento de este conjunto de reglas. Finalmente, se lleva a cabo la acción de la Regla 4, la cual no tiene efecto sobre la información de la BD ya que no hay tuplas cuyo valor del atributo *rate* este entre 1 y 3. El procesamiento de la regla termina dado que no hay una regla que sea disparada por la última acción ejecutada.

3. INTERACCIÓN DE REGLAS

Durante la ejecución de las reglas, estas pueden interactuar de distintas formas. El disparo, la activación/desactivación y conmutatividad de reglas son las formas más comunes de interacción [3]. Para describir e ilustrar estos conceptos se considerará la base de reglas del ejemplo de la Sección 2.

La regla r_i dispara a r_j cuando la ejecución de la acción de la primera genera el evento que dispara a la segunda. La Regla 3

dispara la Regla 2 ya que la ejecución de la acción de la primera origina que se detecte el evento que la segunda regla está supervisando. De igual manera, la Regla 2 se dispara a sí misma ya que su acción coincide con su evento.

La regla r_i activa a r_j siempre que la ejecución de la acción de la primera produzca datos que puedan satisfacer la condición de la segunda. En otras palabras, si la condición de r_j es verdadera después de que r_i ha ejecutado su acción, entonces r_i activa a r_j . La Regla 3 activa a la Regla 2 ya que modifica la información de la BD de manera que existen tuplas que pueden cumplir la condición de la segunda. Sin embargo, la Regla 2 no se activa a sí misma ya que su acción no produce datos que puedan satisfacer su condición.

De manera similar, decimos que r_i desactiva a r_j si la ejecución de la acción de la primera elimina datos que puedan satisfacer la condición de la segunda. En un cierto momento, la Regla 4 podría desactivar a la Regla 3 ya que la acción de la primera elimina todas las tuplas de la relación *account* cuyo valor del atributo *rate* sea mayor a 1 y menor a 3. Como estas tuplas son necesarias para que la condición de la Regla 3 sea verdadera y ya no existen en la relación, la regla queda desactivada.

Finalmente, r_i y r_j conmutan si se obtiene el mismo resultado independientemente del orden en el que se ejecuten sus acciones. La Regla 3 y la Regla 4 no son conmutativas porque si esta última se ejecuta primero puede modificar el comportamiento de la Regla 3.

4. APLICACIONES

Existen diversos sistemas en los cuales las reglas activas juegan un papel preponderante, entre ellas se pueden mencionar las BD activas, la tecnología de casas inteligentes, las aplicaciones financieras y el mantenimiento de restricciones de integridad.

Bases de datos activas

Las BD activas (BDA) combinan la tecnología de BD tradicional con la programación basada en reglas para lograr que la BD reaccione ante ciertos estímulos, posiblemente externos, llamados eventos. La capacidad reactiva de las BDAs es útil para un amplio espectro de aplicaciones, las cuales incluyen seguridad, materialización de vistas, verificación de integridad o integración de BD heterogéneas. Un sistema de BDA consiste de una BD (pasiva) y un conjunto de reglas activas, tal como se muestra en la figura 2.

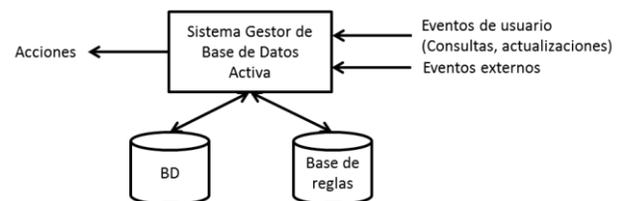


Figura 2. Sistema de base de datos activa

Cuando ocurre un evento relevante dentro o fuera del sistema, el Sistema Gestor de BD activa (SGBDA) lo ejecuta en la BD y también verifica si en la base de reglas existe alguna que responda a tal evento. De ser así, se efectúan los pasos del modelo de ejecución.

Un ejemplo de BDA se muestra en la Sección 2.

Casas inteligentes

Las casas inteligentes usan pequeñas computadoras, distribuidas en la casa, para encender/apagar electrodomésticos o para enviar/recibir información. El uso de las computadoras elimina la necesidad de encender un *switch* o girar una perilla para hacer que las cosas funcionen y permite que los elementos de la casa sean controlados remotamente por, o para responder automáticamente a, las personas que habitan en ella [5]. Esta tecnología es particularmente útil para que los adultos mayores puedan seguir viviendo de manera independiente ya que ofrece una solución viable para ayudarlos en el cuidado de su salud.

La información generada de los sensores dentro de la casa y las conexiones embebidas en aplicaciones domésticas comunes se procesan por un SGBDA para proveer asistencia a las personas de diferentes maneras: prevención de situaciones peligrosas, confort, seguridad y cuidado de la salud.

Prevención de situaciones peligrosas. En estas situaciones la información recolectada del entorno de las personas puede ser verificada y también se pueden tomar en consideración las acciones realizadas previamente. El propósito es proveer una alarma cuando las actividades del habitante lo puedan llevar a situaciones potencialmente peligrosas. Por ejemplo, una persona puede dirigirse de una habitación a otra dentro de la casa y dejar algún electrodoméstico encendido en la habitación de la cual salió. Si la televisión se deja funcionando no es peligroso, pero si se deja la olla de presión y se va a tomar un baño esto sí puede resultar riesgoso y, por lo tanto, se requiere de alguna forma la intervención del sistema. Esto puede representarse con la siguiente regla ECA:

```
ON <salir de la habitación>  
IF <algún aparato está encendido>  
THEN <activar alarma>
```

Confort. Este aspecto se enfoca en el medio ambiente y lo altera de acuerdo a las necesidades de la persona y sus acciones. Por ejemplo, si la temperatura ambiente sube o baja de un cierto límite, entonces hay que reajustarla dentro de la casa.

```
ON <temperatura ambiente por debajo del valor normal>  
IF <hay alguien en la casa>  
THEN <encender el calentador>
```

Seguridad. La gente mayor es vulnerable a ataques de intrusos en su casa. En tales casos, se deben proveer alarmas cuando sea necesario.

```
ON <alarma activada por presencia de un intruso>  
IF <no hay personal esperando en la posición referida>  
THEN <aplica el procedimiento de seguridad>
```

Cuidado de la salud. Aquí la información puede ser accedida desde dispositivos de cuidado de la salud auto-operados, por ejemplo, monitores de presión sanguínea o glucosa. Algunas veces es necesario monitorear signos vitales periódicamente. Por ejemplo, es posible representar con reglas ECA el monitoreo de la presión sanguínea. Si se presenta un incremento sustancial en dos mediciones sucesivas, entonces el personal médico debe ser alertado de esa situación.

```
ON <presión sanguínea mayor a 200/175 durante dos muestras sucesivas>
```

```
IF <el régimen médico para el control de la presión sanguínea ha sido alterado recientemente>
```

```
THEN <notificar al personal médico>
```

Aplicaciones financieras

Un portafolio bursátil se define como una “combinación lineal de instrumentos financieros” [6] tales como bonos. Los administradores de los portafolios se evalúan sobre qué tan bien se comporta su portafolio con respecto a un punto de referencia, el cual está relacionado con medidas numéricas tales como intereses, los cuales se calculan sobre pagos periódicos, y el valor actual del mercado. Además de esas medidas de desempeño, se establecen diferentes restricciones sobre el tipo de inversión permitida para un portafolio dado. La presencia de esas restricciones y medidas de desempeño junto con la naturaleza dinámica del mercado resulta en un portafolio que frecuentemente cambia su composición, tanto en términos de los instrumentos y la cantidad de esos instrumentos presentes en el portafolio. Debido a esto, una BD que apoye eficientemente a las aplicaciones financieras debe proveer apoyo para desarrollar y administrar objetos temporales así como administración del conocimiento, la capacidad de almacenar e imponer reglas que reflejen la semántica de la aplicación a través de reglas temporales y no temporales. Las reglas son el aspecto más importante de la tecnología de BDA para aplicaciones financieras. Facilidades tales como apoyo temporal, tipos de datos abstractos y funciones definidas por el usuario se usan en las reglas para expresar el conocimiento y la semántica de la aplicación. Un ejemplo de una regla aplicada al contexto financiero es la siguiente:

```
ON update stock.price  
IF true  
THEN update-monitor()
```

Mantenimiento de restricciones de integridad

Los sistemas de BD comerciales inicialmente motivaron la introducción de reglas activas (*triggers*) como un mecanismo para programar la integridad referencial entre tablas. Por lo tanto, el mantenimiento de las restricciones de integridad constituye una de las áreas principales de aplicación de las reglas activas.

Las restricciones de integridad pueden clasificarse en estáticas y dinámicas. Las primeras son predicados sobre estados de la BD. Las últimas son predicados sobre transiciones de un estado de la BD a otro. El proceso de verificación de restricciones consiste en evaluar los predicados en orden para determinar su valor de verdad. La verificación de la restricción puede ser inmediata o diferida. La verificación inmediata toma lugar inmediatamente después de la ejecución de una operación que puede violar la restricción; el modo diferido se pospone para el final de la transacción que contiene la operación que puede violar la restricción. Cuando la evaluación de la restricción produce un valor de verdad erróneo, la restricción es violada. El sistema de mantenimiento de integridad reacciona a la violación ya sea “reparando” la restricción, es decir, llevando a cabo cambios a la BD para restaurar el valor de verdad de la restricción, o deshaciendo ya sea el enunciado que causó la violación o la transacción completa. Como ejemplo considérese la siguiente

definición hecha en *Structured Query Language (SQL)* de la relación Empleado(*NombreE, NSS, NSSSuper, ND, Salario*) [7].

```
CREATE TABLE Empleado(  
  NombreE VARCHAR(15) NOT NULL,  
  NSS CHAR(9) NOT NULL,  
  NSSSUPER CHAR(9) NOT NULL,  
  ND INTEGER(1) NOT NULL DEFAULT 1,  
  Salario DECIMAL(10, 2),  
  CONSTRAINT CVEEMP  
    PRIMARY KEY (NSS),  
  CONSTRAINT CVESUPEMP  
    FOREIGN KEY(NSSUPER) REFERENCES  
EMPLEADO(NSS)  
  ON Delete SET Null ON Update Cascade,  
  CONSTRAINT CVEDEPTOEMP  
    FOREIGN KEY(ND) REFERENCES  
DEPARTAMENTO(NUMERO)  
  ON Delete SET Default ON Update Cascade);
```

La instrucción **CREATE TABLE** define un esquema relación de BD, sus atributos y restricciones de integridad. En este caso, se define la relación EMPLEADO con los atributos *NombreE, NSS, NSSSuper, ND* y *Salario*, los cuales almacenan la información del nombre del empleado, su número de seguro social, el número de seguro social de su supervisor, el número de departamento al que está adscrito y su salario, respectivamente. Cada atributo se define junto con su tipo de dato y una indicación de si puede tomar un valor nulo o no.

La palabra reservada **CONSTRAINT** define las restricciones que se deben verificar automáticamente para este esquema. La primera restricción, llamada **CVEEMP**, verifica que el atributo *NSS* sea único para cada empleado. La segunda restricción, **CVESUPEMP**, supervisa los cambios que se hagan al número de seguro social de los empleados que son supervisores. Si se elimina un empleado que es supervisor de otro(s), de forma automática se pone un valor nulo en el atributo *NSSSuper* de todos aquellos empleados que tenían como supervisor al empleado eliminado. En caso de que sólo haya cambiado su número de seguro social, este cambio se propaga al atributo *NSSSuper* de todos los empleados que supervisa. Estas acciones se hacen de forma automática lo que permite mantener fácilmente la consistencia de los datos. La restricción **CVEDEPTOEMP** actúa de manera similar vigilando cuando cambia algún número de departamento para propagar este cambio.

5. CONCLUSIONES

Los sistemas activos son fundamentales para la correcta operación de varios sistemas. El núcleo de estos sistemas son las reglas activas, si estas no están adecuadamente analizadas, diseñadas e implementadas se pueden producir comportamientos inesperados.

6. REFERENCIAS

[1] Paton N. and Díaz O. Active Database System. ACM Computer Surveys, Vol. 31, No. 1, pp. 63-103, 1999.

[2] Paton N., Active Rules in Database Systems, Springer, 1999.

[3] Baralis E. and Widom J. "An Algebraic Approach to Static Analysis of Active Database Rules", ACM Transactions on Database System, Vol. 25, No. 3, pp. 269-332, 2000.

[4] Widom J. and Ceri S., Active Database Systems, Morgan Kaufmann Publishers, 1996.

[5] Augusto J. C., and Nugent C., "A New Architecture for Smart Homes Based on ADB and Temporal Reasoning", In Toward a Human Friendly Assistive Environment (Proc. of 2nd International Conference On Smart homes and health Telematic, ICOST2004), Assistive Technology Research Series, Vol. 14, pp. 106-113, IOS Press, Singapore, September 15-17, 2004.

[6] Chandra R., Segev A., "Active Databases for Financial Applications", in Proceedings of the Fourth International Workshop on Research Issues in Data Engineering, 14 - 15 Feb., pp. 46-52, 1994.

[7] Elmasri R., Navathe S. B., Fundamentos de Sistemas de Bases de Datos, Addison Wesley Iberoamericana, 2002.

[8] Real Academia Española. Disponible en: <http://www.rae.es/>