

A language to express reactivity in the Semantic Web

Elsa Liliana Tovar¹ y María Esther Vidal²

¹Departamento de Computación, Facultad Experimental de Ciencia y Tecnología,
Universidad de Carabobo. Sector Bárbula Naguanagua, Venezuela. Telf: 0241-867.82.43.
eltovar@uc.edu.ve. ²Departamento de Computación y Tecnología de la Información,
Universidad Simón Bolívar. Valle de Sartenejas, Baruta, Venezuela.
Telf: 0212-906.32.68. mvidal@ldc.usb.ve

Abstract

The Semantic Web is a Web to describe by means of metadata the WWW data. In The Semantic Web, knowledge is modeled by using ontology and the reactive behavior is encoded by means of active rules. Hence, active knowledge cannot be used to derive new properties of the data instances, neither new reactive behavior information can be inferred directly from the schema. In this paper, we describe an alternative approach aiming to the enrichment of the Semantic Web with active information. We propose the ontology formalism ACTION. In ACTION, events are categorized as concepts of the ontology and in conjunction with classes, properties and instances are considered during the query answering and reasoning tasks. We have conducted an empirical experimental study and we have observed that active knowledge representation increases the known information in a third of the active knowledge of the modeled domain. But, the augment of the expressive power implies the augment of the computational complexity and efficient strategies to evaluate reactive query are necessary.

Key words: Reactivity, languages of the Semantic Web, knowledge representation, ontologies.

Un lenguaje para expresar reactividad en la Web Semántica

Resumen

La Web Semántica es una Web destinada a describir, a través de metadatos, los datos que están en la WWW. En la Web Semántica, el conocimiento es modelado usando ontologías y el comportamiento reactivo es usualmente codificado independientemente de las ontologías, en reglas activas. Por ello, el conocimiento activo no puede ser usado para derivar nuevas propiedades de los datos, ni nueva información activa. Este trabajo presenta un enfoque alternativo para enriquecer la Web Semántica con conocimiento activo, para ello se propone el formalismo ontológico ACTION. En ACTION los eventos son categorizados como conceptos de la ontología y en conjunción con las clases, propiedades e instancias, son considerados durante las tareas de razonamiento y de responder consultas. Para estudiar la calidad de la solución propuesta, se realizó un estudio experimental. Los resultados del estudio empírico permiten concluir que las ontologías ACTION permiten aumentar hasta en un tercio el conocimiento activo del dominio modelado. Por otra parte, el aumento del poder expresivo ontológico trae consigo un aumento en la complejidad computacional, lo que hace necesaria la definición de estrategias eficientes para evaluar las consultas reactivas que representan las solicitudes de procesamiento reactivo en ACTION.

Palabras clave: Reactividad, lenguajes de la Web Semántica, representación del conocimiento, ontologías.

1. Introducción

La Web Semántica es una *Web* destinada a describir, a través de metadatos (datos acerca de los datos), los recursos que están en la *World Wide Web*. El objetivo de la Web Semántica es crear un espacio donde la información tenga un significado bien definido, de manera que pudiera ser interpretada tanto por humanos como por agentes o aplicaciones de *software*.

Una herramienta útil para dotar de significado los recursos de la *Web* es la ontología. Una ontología es una especificación formal y explícita de conceptos e interrelaciones entre conceptos. En el ámbito computacional, las ontologías han sido usadas exitosamente para modelar datos y conocimiento que permiten conceptualizar dominios particulares; los conceptos reciben el nombre de *clases* y las interrelaciones se denominan *propiedades*. Sin embargo, los esfuerzos por dotar de semántica a la información disponible en la *Web*, se han centrado en una visión estática de los datos o recursos, por lo que las ontologías han sido concebidas básicamente para expresar propiedades estáticas de las entidades relevantes a un dominio de conocimiento. Una propiedad estática es una propiedad que no cambia su valor ante ningún evento, por ejemplo, *nombre* (propiedad) de un *huracán* (clase).

La reactividad de los datos se refiere a la dinámica que rige el comportamiento de los valores que toman los datos cuando se ejecutan las transacciones definidas por el usuario. Esto implica que los datos reaccionan, cambiando automáticamente sus valores, sin la intervención de un programa de usuario. Por ejemplo, la *categoría* (propiedad) de un *huracán* (clase) cambia de acuerdo al *aumento* (evento) de la *velocidad* (propiedad) del viento del *huracán*. En general, el procesamiento de la reactividad es necesario para el control de las restricciones de integridad y el manejo de las reglas del negocio en las bases de datos y, en el caso de la Web Semántica, se ha utilizado extensamente como una herramienta efectiva para modelar esquemas de negociación en comercio electrónico y para el mantenimiento de repositorios XML al prestar servicios electrónicos.

Desde la definición de comportamiento reactivo en los repositorios de datos, el procesamiento de la reactividad se convirtió en un tópico

relevante para la gestión de datos eficiente, dando nacimiento a las bases de datos activas [1]. Las bases de datos activas añadieron una nueva dimensión de independencia de los datos, la llamada independencia del conocimiento activo. Esta independencia libera a las aplicaciones del procesamiento de los cambios a los datos como consecuencia de los eventos o transacciones que los afectan y delega esta gestión a la ejecución de las reglas activas bajo el paradigma Evento-Condicción-Acción (ECA) [1]. Este enfoque no facilita la incorporación del comportamiento reactivo de los datos en la construcción de las ontologías de la Web Semántica, sino que por el contrario, establece una clara separación entre el conocimiento activo codificado en las reglas y la conceptualización del dominio modelado en la ontología y los eventos no son conceptos sino transacciones como inserción, eliminación, actualización, etc.

En este trabajo, se presenta un formalismo para definir ontologías activas con el propósito de enriquecer a la Web Semántica con la posibilidad de expresar comportamiento reactivo. En nuestro formalismo, los eventos son considerados como conceptos al mismo nivel que las clases de la ontología y, en conjunción con las clases, propiedades y las instancias, son tomados en cuenta durante las tareas de razonamiento y de responder consultas. Este artículo consiste de siete secciones. En la sección 2, se describe un ejemplo que motiva la importancia de representar conocimiento reactivo en la Web Semántica. En la sección 3, se describen los trabajos relacionados con el tema. En la sección 4 se describe el formalismo propuesto. En la sección 5 se describe el procesamiento del comportamiento reactivo. En la sección 6 se reportan los resultados de nuestro estudio experimental. Finalmente, en la sección 7 se presentan las conclusiones y los trabajos futuros.

2. Motivación

Considere que el dominio de los desastres naturales se modela a través de una ontología y se conoce que: todo fenómeno natural es un fenómeno causado por la naturaleza, estos pueden ser: terremotos, huracanes, erupciones volcánicas, etc. Cada uno de ellos tiene características que le son propias, por ejemplo: un terremoto tiene una onda sísmica cuya amplitud expresa su

poder destructivo, así como la velocidad del viento es la principal propiedad de un huracán. Todos estos conceptos pueden ser modelados con lenguajes que permiten definir esquemas para los datos en ontologías básicas tales como RDF/RDFS [2]. Por ejemplo, se puede definir que *FenómenoNatural*, *Terremoto*, *Huracán*, *Tifón* son clases del dominio y que la *intensidad* es una propiedad de *Terremoto* y *categoría* es una propiedad de *Huracán*. Además, es posible definir que las clases *Huracán* y *Terremoto* son subconjuntos de *FenómenoNatural*. No obstante, sólo con un lenguaje como OWL [3], que contiene el constructor *equivalentClass*, es posible modelar que las clases *Huracán* y *Tifón* son sinónimos. La información acerca de que *terremotoIndonesia2004* es un individuo que pertenece a la clase *Terremoto* y que el valor de su propiedad *intensidad* es 7.1, constituye, en conjunción con la información anterior, el conocimiento estático debido a que ninguno de estos datos es afectado por la ocurrencia de ningún evento. Sin embargo, hay otros aspectos del dominio que necesitan ser representados. Por ejemplo, las lluvias abundantes durante un período de tiempo largo pueden ocasionar deslizamientos, los terremotos pueden activar tsunamis si el lecho marino se quiebra o puede causar daños a las estructuras civiles por incendios o inundaciones. En estos casos, un sistema de *software* activo, que trabaje con este dominio, puede reconocer estas situaciones y mandar, de forma automática, mensajes de alerta como *alertaTsunami*, *alertaInundación* o *alertaIncendio* a los organismos competentes y, probablemente así, evitar tragedias mayores.

Con los formalismos existentes es posible modelar en la ontología, una propiedad llamada *alertaGenerada* cuyos posibles valores sean *alertaTsunami*, *alertaInundación* y *alertaIncendio*. Sin embargo, no es posible expresar que el *alertaTsunami* sólo debe ser enviado cuando el lecho marino se ha deformado por causa del terremoto, es decir, cuando otra propiedad como, por ejemplo, *daño_Causado* por el terremoto tiene el valor "*lecho_deformado*". Esta limitación se evidencia también cuando se quiere expresar que los únicos valores posibles para una propiedad como *categoría* de la clase *Huracán* son 1, 2, 3, 4 y 5 y que estos valores son determinados por el *incremento/decrecimiento* de la *velocidad* del viento (la categoría de un huracán es 1 si la velocidad es ma-

yor a 153 Km./hora, 2 cuando es mayor 177 Km./hora y así). Sólo con una visión global de los cambios de los datos es posible modelar información estática y activa de este dominio. Sin embargo, esta visión no puede ser expresada con las tradicionales clases y propiedades de los formalismos existentes. Se requiere considerar a los eventos como conceptos y a ciertas propiedades como propiedades activas. En este trabajo, proponemos un formalismo para ontologías activas a fin de representar conocimiento estático. El formalismo se implementa como una base de datos deductiva, la cual permite derivar nuevo conocimiento sobre las propiedades activas a partir de la aparición de eventos. Tanto el conocimiento derivado a partir de la existencia de un evento, así como el conocimiento estático, pueden ser considerados indistintamente durante las tareas de razonamiento y de responder consultas.

3. Trabajos Relacionados

Las bases de datos activas han sido gestionadas con un *software* que integró el manejo tradicional y eficiente de las bases de datos tradicionales con la creación y ejecución de reglas de producción. Manejadores de bases de datos activas tradicionales como Starburst, Oracle, DB2 y Stalog hicieron uso de *triggers* de propósito general basado en el estándar de SQL3 [1]. En el ámbito de las bases de datos orientadas a objeto se presenta el proyecto CHIMERA [4] que hace uso de un lenguaje de reglas activas propio y utiliza el identificador de objetos para enlazarlo a los eventos que los afectan. En dicho proyecto se plantea, a nivel de diseño conceptual, un nuevo diagrama llamado "*Evolución de los Objetos*" para mostrar cómo deben cambiar los datos cuando ocurren los eventos.

En el contexto de la *Web Semántica* también se ha incorporado el paradigma de las reglas ECA para gestionar reactividad [5-10]. En [6] se especifican reglas ECA, en el contexto de XSLT [11] y LOREL [12], para implementar servicios y comercio electrónicos haciendo posible generar o manipular el contenido de un repositorio XML como una reacción a los cambios que ocurren en la información de los documentos. En [13] se presenta un trabajo lógico para modelar cambios sobre una ontología a través del tiempo. Este enfoque consiste en un marco de trabajo ontológico para representar actividades, estados y tiempo se

representan dentro de una ontología en lógica de primer orden. Un agente inteligente es definido para razonar acerca de la ocurrencia de acciones que modifican a los datos. No se define el concepto *evento* y el conocimiento representado en la ontología es estático y sólo cambia si el agente externo detecta que después de pasado un tiempo se deben ejecutar las acciones. Recientemente, se presentó en [14] un enfoque para integrar información basada en ontologías a fin de expresar en ella los cambios frecuentes de ciertos fragmentos de metadata acerca de fuentes de datos heterogéneas. Sin embargo, el objetivo de ese trabajo no apunta hacia representar conocimiento reactivo. En [9, 10] se define el lenguaje *XChange* para modelar las reglas ECA introduciendo un álgebra de composición de eventos que permite definir eventos compuestos y simples que pueden ocurrir en distintos sitios en la *Web*. A diferencia de la presente propuesta todos estos trabajos se centran en la implementación de las reglas ECA como paradigma de representación del comportamiento reactivo, por esa razón los eventos (ya sean compuestos o simples) son vistos, exclusivamente, como transacciones sobre los datos.

En [15, 16], se plantea la necesidad de procesar reactividad en la *Web Semántica* y cómo los eventos en la *Web* no deben estar restringidos al enfoque de las bases de datos activas donde sólo se consideren inserciones, actualizaciones, eliminaciones, etc. Debido a que un evento en la *Web* puede generarse por cualquier ocurrencia externa e incontrolada que puede resultar en la realización de un conjunto de acciones [15], se maneja un enfoque de los eventos como sistemas reactivos. Con esa visión más compleja de la reactividad, las reglas ECA son usadas, en su semántica general, para representar un paradigma reactivo a ser modelado a través de otros tipos de reglas, como por ejemplo: las reglas deductivas, las reglas declarativas, otras reglas de producción, etc. Básicamente, se parte de la idea de reglas abstractas [16] que tienen diferentes componentes. Los componentes de las reglas abstractas son descritos a través de una ontología llamada r^3 [16]. La ontología r^3 especifica el vocabulario abstracto de los elementos constitutivos de un lenguaje de reglas y cuál es la función de cada uno de ellos. Como continuación del trabajo sobre r^3 , se definió en [15] un formalismo para representar políticas de tratamiento de la reactivi-

dad en la *Web* de forma tal que los agentes pueden automáticamente ejecutar tareas como resultados de eventos. Los aportes de [15, 16] están claramente circunscritos al ámbito del uso de las reglas ECA como mecanismo para procesar reactividad. En este caso la ontología r^3 representa información estática de los elementos que componen a las reglas ECA, no expresan información activa de ningún dominio a modelar. Por esta razón carecen de la posibilidad de usar el conocimiento activo en conjunción con los datos durante el procesamiento de consultas y las tareas de razonamiento propias de las bases de conocimiento ontológicas a diferencia del presente trabajo.

4. Formalización de Ontologías Activas: Formalismo ACTION

En este trabajo se propone una *ontología activa* para expresar conocimiento activo y estático de acuerdo a las siguientes definiciones:

Definición 1

(ACTION: Ontología activa O_a)

Una ontología activa es una 7-tupla $O_a = \langle C, E, Pe, Pa, F, A, fr \rangle$ donde:

- C: un conjunto de clases y tipos de datos básicos.
- E: un conjunto de eventos.
- Pe: un conjunto de propiedades estáticas; cada propiedad corresponde a una función que va de $C \cup E$ en $C \cup E$.
- Pa: un conjunto de propiedades activas; cada propiedad activa corresponde a una función que va de C en C.
- F: un conjunto de predicados que representan instancias de las clases, propiedades y eventos.
- A: un conjunto de axiomas que describen las propiedades con semántica predefinida provistas en Pe y en Pa.
- fr es una función, fr: $F \times \text{Booleano}(Pa \cup Pe) \times E \rightarrow F$; fr define el comportamiento reactivo en O_a , donde $\text{Booleano}(Pa \cup Pe)$ corresponde al conjunto de expresiones Booleanas que se pueden construir sobre las propiedades activas en Pa y las propiedades estáticas en Pe.

Retomando el ejemplo de la sección 2, la función fr define el comportamiento de la propiedad activa $categoria$ del huracán ike , que su cambia su valor a 5 cuando ocurre el evento $incrementoVelViento$ y la propiedad $velViento$ es mayor a 250 km/hora, denotado como $fr((ike,categoria,4),(categoria,event(incrementoVelViento,velViento \geq 250)) \rightarrow (ike,categoria,5)$

El conjunto de propiedades Pe contiene propiedades que inducen una jerarquía de eventos y una jerarquía de clases. De esta forma, se puede representar en una ontología activa Oa que un evento $afectoLechoMarino$ es un subevento del evento $afectoVida$ a través del hecho $isSubEventOf(afectoLechoMarino,afectoVida)$, en el conjunto F de Oa , tal como se puede observar en la Figura 1. Cabe destacar que la condición booleana, que dispara las modificaciones a las propiedades activas cuando ocurren los eventos, son condiciones sobre los datos (otras propiedades activas o pasivas) y nunca una operación que deba ser realizada para luego ser comparada en la expresión como, por ejemplo, el cálculo de un promedio o de una totalización sobre los valores de otra propiedad. Por otra parte, en el conjunto de axiomas A de la ontología activa Oa debe aparecer una regla deductiva que exprese que el predicado con semántica predefinida $isSubEventOf$ es transitivo.

La ontología ACTION definida representa la forma canónica de un tipo de ontología activa que consta de: 1) un componente extensional: los predicados que representan información explícita de los datos como $isSubEventOf(afectoLechoMarino,afectoVida)$, y 2) un componente intensional:

que define la semántica de los predicados extensionales, por ejemplo, "si $E1$ es un subevento de $E2$ y $E2$ es un subevento de $E3$ entonces $E1$ es un subevento de $E3$ ". Debido a estas componentes, una ontología ACTION se puede implementar como una base ontológica deductiva. En [17] se han desarrollado técnicas eficientes de optimización y evaluación de consultas y optimización de tareas de razonamiento en bases ontológicas deductivas. Este enfoque es extendido en el presente trabajo añadiendo predicados meta-level para representar comportamiento reactivo de los datos. Según esto, una ontología activa se puede modelar de la siguiente forma:

Definición 2
(Base Ontológica Deductiva Activa)

Dada una ontología activa $Oa=(C,E,Pe,Pa,F,A,fr)$, una Base Ontológica Deductiva Activa ADOB, para Oa , es un par $\langle AEO,AIO \rangle$, donde:

- AEO corresponde a base activa extensional compuesta por meta-level predicados que representan explícitamente el conocimiento activo presente en los conjuntos C, E, Pe, Pa, F ;
- AIO corresponde a una base activa intensional compuesta por meta-level predicados que definen la semántica del conocimiento representado en AEO y modelado por los axiomas que están en A .

Cada ADOB se compone de predicados meta-level como, por ejemplo, los predicados de semántica predefinida $isEvent(E)$, donde E es el nombre de un evento, y $isSubEventOf(E_1,E_2)$, donde E_1 y E_2 son nombres de eventos y se define

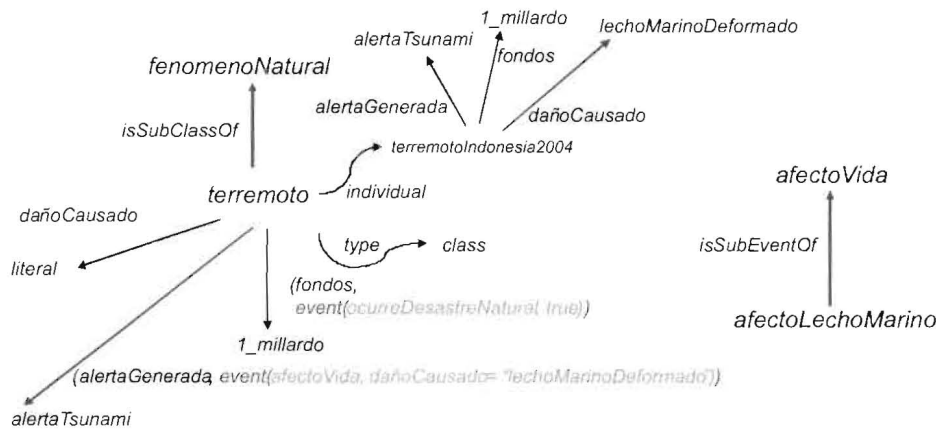


Figura 1. Representación de una ontología activa Oa .

que define que E_1 es un subevento de E_2 . Un ejemplo de estos predicados sería *isEvent(afectoLechoMarino)* que representa un hecho que forma parte del componente extensional AEO, así como también, *isSubEventOf(afectoLechoMarino,afectoVida)*. Por otra parte, la componente intensional AIO está compuesta de predicado intensionales como *areSubEvents(E_1, E_2)*. La evaluación de *areSubEvents(E_1, E_2)* define la clausura transitiva del predicado *isSubEventOf(E_1, E_2)*. Otros predicados de AEO como el predicado *activeProperty(P, T, D, R)* definen una propiedad activa AP en términos de su tipo T , dominio D y rango R ; y el predicado *reactiveBehavior(AP, E_1, P, V, RV)* especifica el comportamiento reactivo de una propiedad AP que toma el valor RV cuando un evento E_1 ocurre y la propiedad P tiene el valor V . Por ejemplo, *reactiveBehavior(categoría, incrementoVelViento, velViento, 251, 5)* que indica que la propiedad activa *categoría* debe cambiar a 5 si ocurre el evento *incrementoVelViento* y *velViento* se ubica en 251. La interpretación de las variables de ADOB sobre la estructura I es una función μ que hace corresponder los nombres de las variables de V_v en elementos del un dominio de interpretación D .

Definición 3 (Modelo Mínimo de ADOB)

El modelo mínimo MM de una ontología ADOB corresponde con el modelo para el cual no existe ningún otro modelo MM' para ADOB tal que $MM' \subset MM$ ([1]). El MM es el punto fijo sobre el conjunto de datos que se alcanza cuando se aplican las reglas de la base de datos intensional AIO a partir de los hechos de la base de datos extensional AEO.

El procesamiento de la reactividad de una ontología $ADOB_1$ cuando ocurre un evento e , se define como el modelo mínimo de $ADOB_1 U \{isEvent(e)\}$. Intuitivamente, el modelo mínimo está formado por todos los hechos que pueden ser deducidos o derivados a partir del conocimiento explícitamente expresado en la ontología, los axiomas y las propiedades activas derivadas a partir del evento e que se disparó.

5. Procesamiento del Comportamiento Reactivo

El procesamiento del comportamiento reactivo implica una evaluación *bottom-up semi-*

naive [1] de un conjunto de reglas como la siguiente:

```
areSupEvents(F, E) :- isSubEventOf(E, F).
areSupEvents(F, E) :- isSubEventOf(E, G),
areSupEvents(F, G).
```

En el procesamiento de la reactividad generada por el evento E , supongamos que el evento es *incrementoVelViento*, se determinan todos los individuos afectados por ese evento, es decir se evalúa el predicado *areIndividuals(I, Huracán)* que retorna todos los huracanes activos. Luego, las propiedades activas de los huracanes –por ejemplo *categoría*– que están afectadas por *incrementoVelViento*, cambian de acuerdo a la especificación del comportamiento reactivo codificado en el predicado *reactiveBehavior(categoría, incrementoVelViento, velViento, 251, 5)*. Se modifican las propiedades activas –la *categoría* de cada huracán– y se procede igual para todos los supereventos de E . El algoritmo que procesa la reactividad de este modo tiene un tiempo polinomial debido a que el costo de la clausura transitiva está acotado por n^3 [18], donde n es el número de predicados *isSupEventOf*. Si M es el número de individuos con propiedades activas a modificar entonces se puede decir que el orden de este algoritmo es $O(n^3 * M)$.

6. Resultados Experimentales

Para estudiar empíricamente la capacidad expresiva del formalismo, se realizó un estudio experimental en base al *Lehigh University Benchmark (LUBM)* [19] considerado un punto de referencia estándar para evaluar razonamiento. Este *benchmark* en su versión original genera ontologías sintetizadas con información estática. En este trabajo se hizo una extensión del generador para que las ontologías constaran de eventos y propiedades activas como parte del dominio representado. La plataforma utilizada fue una máquina Solaris con procesador Sparcv9 1281 MHz y 16GB de memoria RAM y el lenguaje usado para el razonador fue SWI-Prolog, Version 5.6.54. La primera parte del estudio experimental tuvo el propósito de medir la relación entre conocimiento implícito y el conocimiento explícito aumentando progresivamente el conocimiento explícito (número de instancias del ABOX). Se calculó el modelo mínimo (MM) haciendo uso de un meta-

intérprete en cada caso. La hipótesis inicial sugería que el procesamiento reactivo de una ontología genera un MM mayor en la medida en que el número de instancias de propiedades activas a modificar es mayor. La métrica utilizada fue N_{MM} , que representa el número de hechos derivados en el modelo mínimo calculado. En la Tabla 1 se muestran las características de las ontologías utilizadas y el tamaño del N_{MM} obtenido. Los resultados confirman un crecimiento del N_{MM} polinomial respecto al número de instancias generadas en cada caso.

La segunda parte del estudio consistió en medir el impacto de la representación del conocimiento activo de nuestro formalismo en relación al conocimiento inferido. La hipótesis inicial sugería que la representación del conocimiento activo en nuestro formalismo permitía inferir nuevo conocimiento en relación al número de eventos que afectan a los datos. Para probar esta hipótesis, se ejecutó el procesamiento reactivo midiendo el aumento del número de hechos inferidos.

La Figura 2 consta de dos gráficas, la gráfica 2-a muestra cómo el conocimiento activo inferido puede crecer repentinamente al ocurrir un evento que tenga muchos supereventos o no crecer de manera proporcional cuando se procesa un evento que no tiene supereventos en la jerar-

quía. La gráfica 2-b muestra el estudio de regresión lineal aplicado a los puntos de cada curva, independientemente del tamaño de la jerarquía. En ella se observa el coeficiente de determinación (R^2), que mide la proporción de la variación del número de hechos inferidos que puede ser explicada por la variación del número de eventos ocurridos. La variación total del aumento del nuevo conocimiento activo es de 19.5% al aumentar de 1 a 5 el número de eventos que afectan a cada propiedad activa. Considerando que variar hasta 5 el número de eventos implica añadir hechos que representan sólo el 0.39% del total de hechos de la base de datos extensional y que eso genera un aumento del conocimiento activo en el orden del 19.5%, permite concluir que la capacidad expresiva del formalismo propuesto permite inferir una cantidad importante de nuevo conocimiento activo al procesar el conjunto de eventos que ocurren en un dominio de conocimiento modelado el formalismo ACTION.

7. Conclusiones y Trabajos Futuros

Todo el trabajo de investigación y la experimentación empírica nos lleva a las siguientes conclusiones: la semántica del conocimiento reactivo codificada en las reglas ECA está separa-

Tabla 1
Características de las ontologías utilizadas en el estudio experimental

	# Univ	# Inst Clases	# Inst Prop	# Inst Act Prop	MB	NMM
Pequeña	1	15393	55528	11802	5,7	455624
Mediana	5	91408	325429	69441	33,5	204703
Grande	10	184086	652868	139438	67,3	3796888

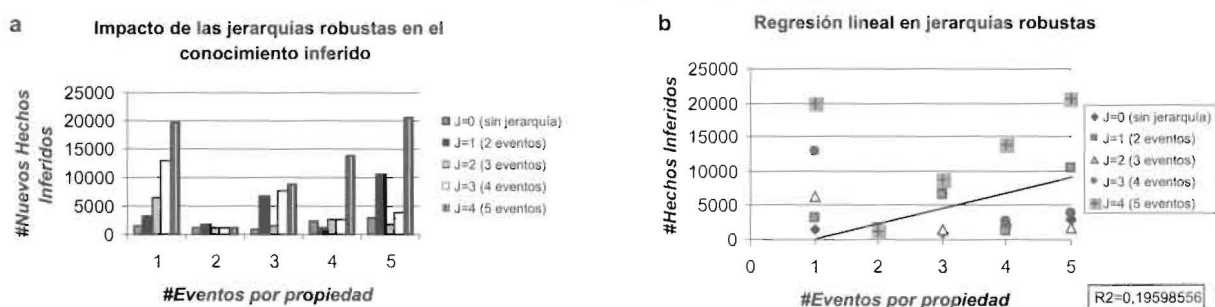


Figura 2. a) Conocimiento activo derivado por la ocurrencia de diferentes eventos sobre una propiedad activa variando el tamaño de la jerarquía. b) Curva de regresión lineal sobre el conjunto de puntos generados al procesar la reactividad.

da de la semántica de los datos en el esquema, por lo que el esquema de los datos no expresa su reactividad y no se pueda inferir nueva conocimiento reactivo. Adicionalmente, el enfoque centrado en los eventos como transacciones sobre los datos restringe la administración de la reactividad a un enfoque exclusivamente procedimental y no se explota, en forma alguna, el uso de los metadatos para inferir conocimiento reactivo. Esto nos lleva a observar que las reglas ECA no escalan para añadir poder expresivo reactivo a los lenguajes de la Web Semántica. Por otra parte, la posibilidad de expresar eventos como conceptos al mismo nivel que las clases en las ontologías activas, permite hacer un uso efectivo de mecanismos de inferencia para procesar comportamiento reactivo a diferencia del uso del paradigma de las reglas ECA. Esta característica abre un nuevo escenario en el manejo de la reactividad en la Web Semántica. El concepto evento permite establecer nuevas relaciones de contención de conceptos, para crear jerarquías de generalización entre eventos, favoreciendo una mejor descripción del dominio de interés y, adicionalmente, los datos y metadatos acerca de los eventos, las clases, las propiedades y las instancias pueden ser usados conjuntamente para resolver tareas de razonamiento y de responder consultas. Debido a que existe una relación inversamente proporcional entre el poder expresivo de un lenguaje y su complejidad computacional, añadir la capacidad de expresar reactividad a los lenguajes ontológicos hace más costosas las tareas de razonamiento y de responder consultas. Por esta razón, un trabajo futuro consiste en buscar estrategias eficientes para evaluar consultas que representen el procesamiento del comportamiento reactivo de los datos en ADOB, los resultados preliminares de esta propuesta se reportan en [20].

Referencias Bibliográficas

1. Zaniolo C, et al: "Advanced Database Systems". Morgan Publishers, USA (1997), 5-33.
2. Hayes P.: "RDF Semantics: W3C Working Draft". (2003). <http://www.w3.org/TR/rdf-mt/>
3. McGuinness D, van Harmelen F.: "OWL, Recommendation W3C", (2004). <http://www.w3.org/TR/owl-features>.
4. Foster I, Voekler J., Wilde M., Zhao Y.: "Chimera: A virtual data system for representing, querying and automating data derivation". In Proceedings of the 14th SSDBM, (1995). 37-46.
5. Bailey J., Poulouvasilis A., Wood P. "Analysis and optimization for event-condition-action rules on XML". Computer Networks, April, Vol. 39, (2002). 239-259
6. Bonifati A, et al: "Active rules for XML: A new paradigm for e-services". VLDB Journal Vol. 10, (2001). 39-47.
7. Papamarkos G., Poulouvasilis A., Wood P.: "Event-Condition-Action Rule Languages for the Semantic Web". In Proc. WWW'2002, (2002) Hawai USA. 486-495.
8. Grigoris A. et al. "Reasoning Methods for Personalization on the Semantic Web". Annals of Mathematics, Computing & Teleinformatics, Vol. 2, no 1, 2004, pp 1-24.
9. Bry F. et al. "Reactivity on the web: paradigms and applications of the language Xchange". Journal of Web Engineering, Vol. 5, No. 1 (2006) 003-024.
10. Bry F., Eckert M., Patranjan P. "Querying Composite Events for Reactivity on the Web". LNCS, Vol. 3842, International Workshop on XML Research and Applications (XRA 2006), pp 38-47.
11. Berglund A.: "XSLT Specification". (1999). www.w3.org/TR/WD-xsl
12. Abiteboul S., Quass D., McHugh J., Widom J., Wiener J.: "The Lorel Query Language for Semistructured Data". International Journal on Digital Libraries. April (1997).
13. Gruninger M, Fox M.: "An Activity for Enterprise Modeling" (2006). www.eil.utoronto.ca/enterprise-modelling/papers/gruninger-wetice94-act.pdf.
14. Xing W., Corcho O., Goble C., Dikaiakos M.: "Active Ontology: An Information Integration Approach for Dynamic Information Sources". Poster at 4th European Semantic Web Conference (2007). <http://grid.ucy.ac.cy/Talks/ActOn-xing.UCY.ppt>
15. May M., Alferes J, Amador R. "An Ontology- and Resources-Based Approach to Evolution and Reactivity in the Semantic Web". LNCS

- Vol. 3761, Active Rules and Web Services 2005 pp. 1553-1570.
16. Alferes J, Amador R., Kärger P., Olmedilla D. "Towards reactive semantic web policies: Advanced agent control for the semantic web". In International Semantic Web Conference (Posters & Demos), 2008. <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-401/>
 17. Ruckhaus E., Ruiz E., Vidal M.E.: "Query Optimization in The Semantic Web". International Workshop on Applications of Logic Programming in the Semantic Web and Semantic Web Services. Seattle, USA (2006). http://ldc.usb.vt/~ruckhaus/es/investigación/assets/rrv_cr.pdf
 18. Cohen E.: "Size-estimation framework with applications to transitive closure and reachability". ACM Journal of Computer and System Science, Vol 55 Academic Press, FL, USA, (1997) 441-453.
 19. SWAT Projects, "The Lehigh University Benchmark (LUBM)". <http://swat.cse.lehigh.edu/projects/lubm/>
 20. Tovar E., Vidal M.E.: "Magic Rewritings for Efficiently Processing Reactivity on Web Ontologies". LNCS Vol 5332. Proceedings of the OTM 2008. 1338-1354.

Recibido el 9 de Marzo de 2009

En forma revisada el 22 de Febrero de 2010