



Universidad Simón Bolívar
Doctorado en Ciencia de la Computación

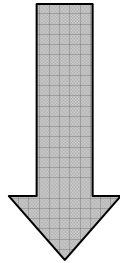
Representando Reactividad en la Web Semántica

Elsa Liliana Tovar

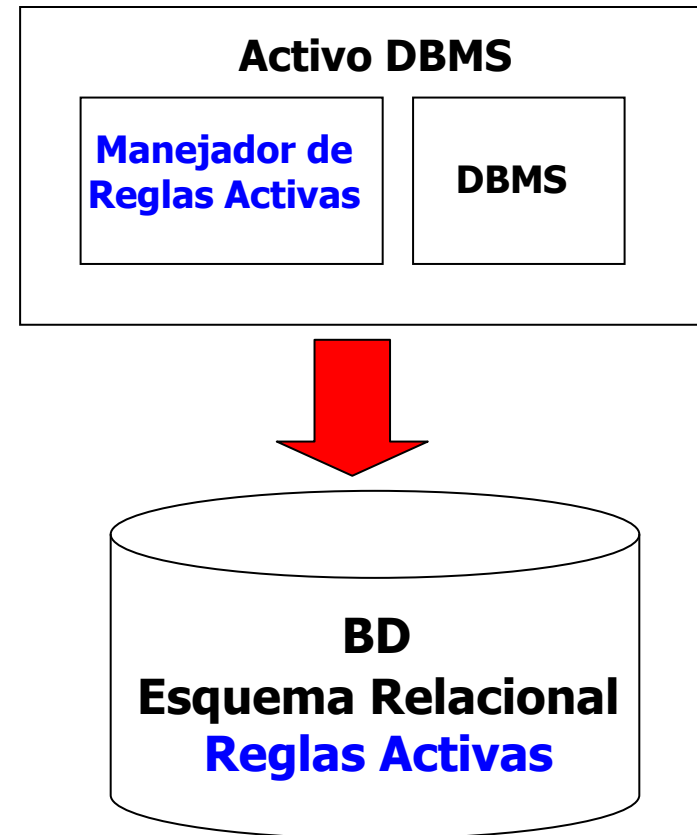
Febrero, 2008

Reactividad

- Reglas de Producción: *paradigma Evento-Condición-Acción*
- Independencia del Conocimiento

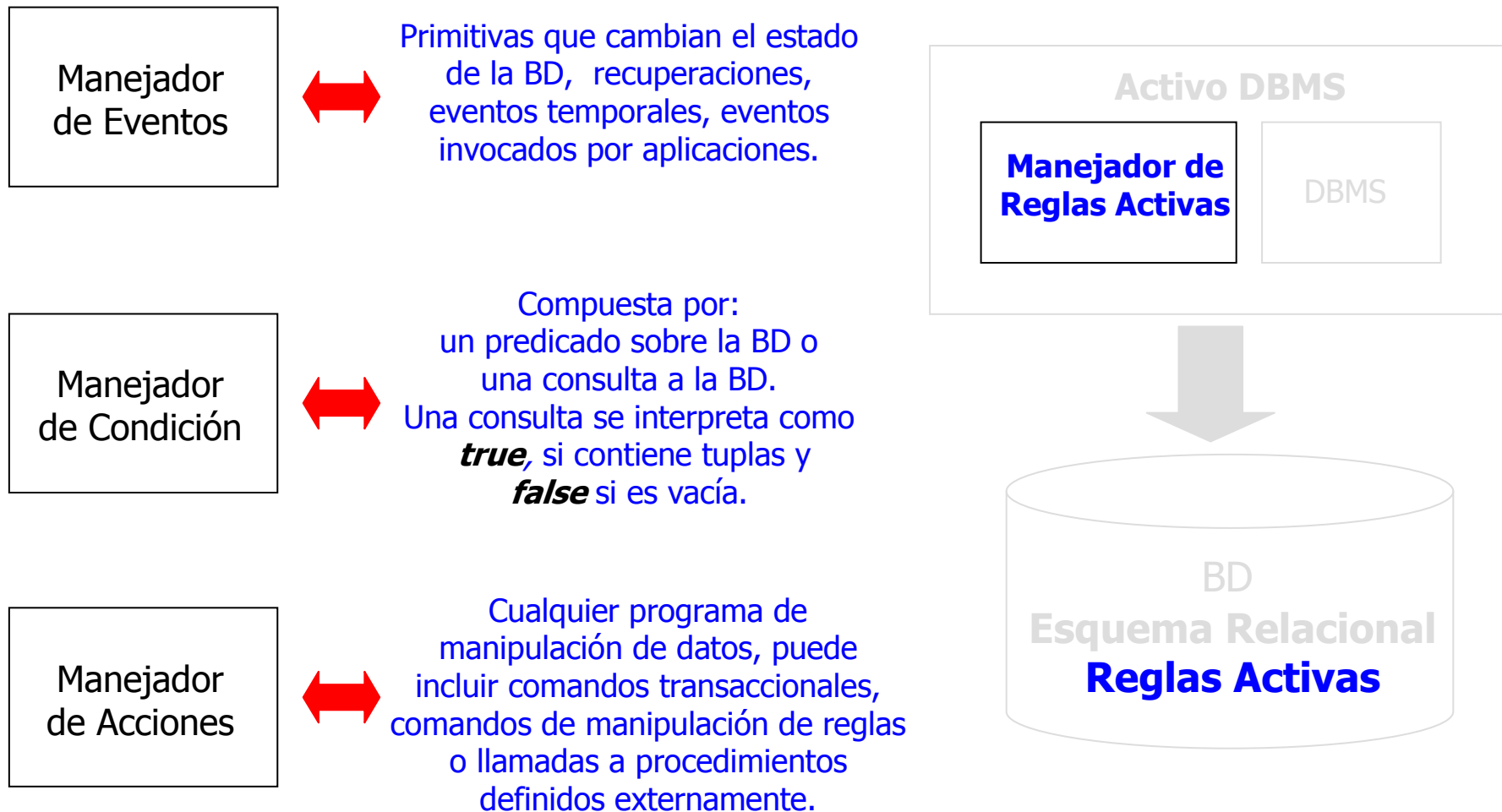


Aplic. Internas como integridad, mantenimiento de vistas
Aplic. Externas como procesamiento de las reglas del negocio.



Sintaxis General

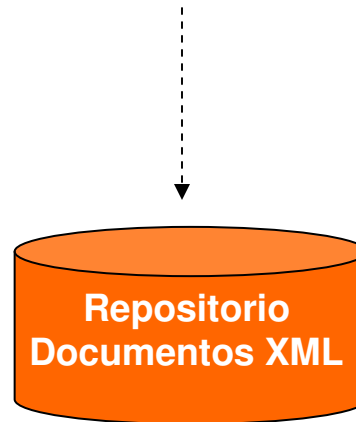
on event if condition do actions



Reglas ECA en la WS

Aplicaciones: e-services, e-commerce

**Lenguaje de Reglas ECA
para soportar funcionalidad reactiva
*prototipo***

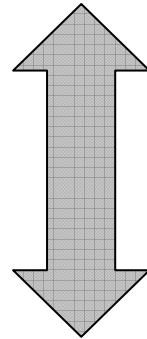


XML ECA Rule Language

on *event* **if** *condition* **do** *actions*

XPath:

Selección de fragmentos del documento en XML



XQuery:

Inserciones donde se deben construir nuevos fragmentos en XML



Ejemplo: los.xml

```
<LOs>
  <LO type="book" title="Data On the Web">
    <subject>Computer Science</subject>
    <creator>S. Abiteboul</creator>
    <creator>P. Buneman</creator>
    <creator>D. Suciu</creator>
    <description>From Relations to Semistructured data and XML
    </description>
    <publisher>Morgan Kaufmann</publisher>
    <isbn>1-55860-621-Y</isbn>
    <annotations>
      <review>
        <reviewer>Teacher Education Review Panel</reviewer>
        <date>2002-10-20</date>
        <rating>9</rating>
        <description>
          This book gives a comprehensive, state-of-the art
          discussion of data models, query languages and ...
        </description>
      </review>
      <review>
        <reviewer>John Smith</reviewer>
        <date>2002-12-20</date>
        <rating>10</rating>
        <description>
          I found this a great book to learn about querying
          semi-structured data, which I didn't know much about.
        </description>
      </review>
    </annotations>
  </LO>
  ...
</LOs>
```

Ejemplo: users.xml

```
<users>
  ...
  <user id="217">
    <name>Johnny Mnemonic</name>
    <profession>student</profession>
    <subjects>
      <subject>Computer Science</subject>
      <subject>Mathematics</subject>
      <subject>Economics</subject>
    </subjects>
    <LOs>
      <LO type="book" title="Data On the Web">
        <isbn>1-55860-621-Y</isbn>
        <latest-review>
          <reviewer>John Smith</reviewer>
          <date>2002-12-20</date>
          <rating>10</rating>
          <description>
            I found this a great book to learn about querying
            semi-structured data, which I didn't know much about.
          </description>
        </latest-review>
      </LO>
      ...
    </LOs>
  </user> ...
</users>
```

Reglas XML ECA

```
ON INSERT document('los.xml')/LOs/LO/annotations/review
IF $delta/../../../../subject[.='Computer Science']
DO DELETE document('users.xml')/users/user[@id="217"]/LOs/LO
    [isbn=$delta/../../../../isbn]/latest-review;
INSERT <latest-review>{$delta/*}</latest-review>
BELOW document('users.xml')/users/user[@id="217"]/LOs/
    LO[isbn=$delta/../../../../isbn]
AFTER isbn

INSERT <review>
    <reviewer>Neo Anderson</reviewer>
    <date>2003-04-29</date>
    <rating>9</rating>
    <description>
        Very clearly written and very well-organised.
        Describes in detail all the ...
    </description>
</review>
BELOW document('los.xml')/LOs/
    LO[isbn="1-55860-621-Y"]/annotations
AFTER TRUE
```

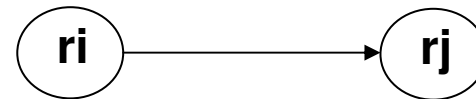
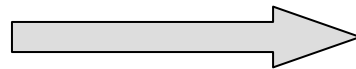
Triggering Graph para XML

Técnica: creación del grafo a través de prueba de contención entre expresiones *path*

ri:	ON DELETE j	evento
	IF \$delta\....\ loquesea1.xml	condición
	INSERT r BELOW e BEFORE q	acción

rj:	ON INSERT s	evento
	IF \$delta\....\ loquesea2.xml	condición
	INSERT l BELOW i BEFORE k	acción

$s = e \setminus r$



Enfoque Conservador: sólo los grafos acíclicos garantizan *termination*.

Limitaciones de las reglas ECA en la WS

- El enfoque centrado en los **eventos como *transacciones*** sobre los datos **restringe** la administración de la reactividad a un **enfoque exclusivamente procedimental**.
- La **semántica del conocimiento reactivo codificada en las reglas ECA** está separada 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.
- No se explota, en ninguna forma, el uso de los metadatos para inferir conocimiento reactivo.
- Las reglas ECA **no escalan para añadir poder expresivo reactivo a los lenguajes de la Web Semántica**.

```

ON INSERT document('los.xml')/LOs/LO/annotations/review
IF $delta/../../subject[.='Computer Science']
DO DELETE document('users.xml')/users/user[@id="217"]/LOs/LO
  [isbn=$delta/../../isbn]/latest-review;
INSERT <latest-review>{$delta/*}</latest-review>
BELOW document('users.xml')/users/user[@id="217"]/LOs/
  LO[isbn=$delta/../../isbn]
AFTER isbn

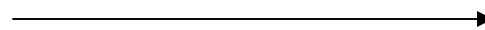
```

```

INSERT <review>
  <reviewer>Neo Anderson</reviewer>
  <date>2003-04-29</date>
  <rating>9</rating>
  <description>
    Very clearly written and very well-organised.
    Describes in detail all the ...
  </description>
</review>
BELOW document('los.xml')/LOs/
  LO[isbn="1-55860-621-Y"]/annotations
AFTER TRUE

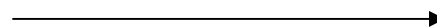
```

R1:
on insert review
 subject="Computer Science"



metadata de
 Johnny Mnemonic

R2:
on insert review
 subject="Computer Science" and
 contributor= "scientist"

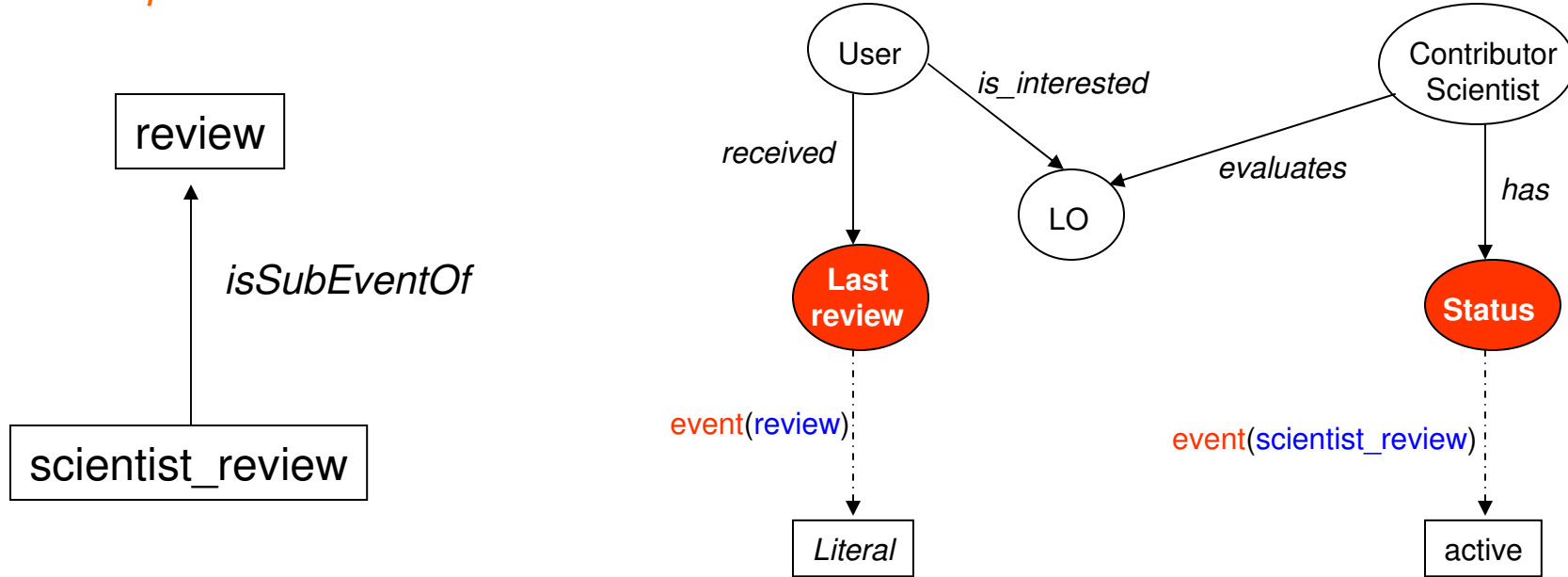


metadata de
 Neo Anderson en
 Contributor_Scientist

Desventajas:

- Por cada funcionalidad reactiva en la aplicación, se debe crear una regla nueva.
- Requiere de un esquema de ejecución de reglas que garantice terminación.
- No se pueden establecer relaciones entre los eventos de dos o más reglas, aún si estos son el mismo.
- La semántica del comportamiento reactivo codificada en las reglas es transparente para el manejador de la reactividad.

Jerarquía de eventos



“Si ocurre un sub-evento ocurren todos sus ancestros”

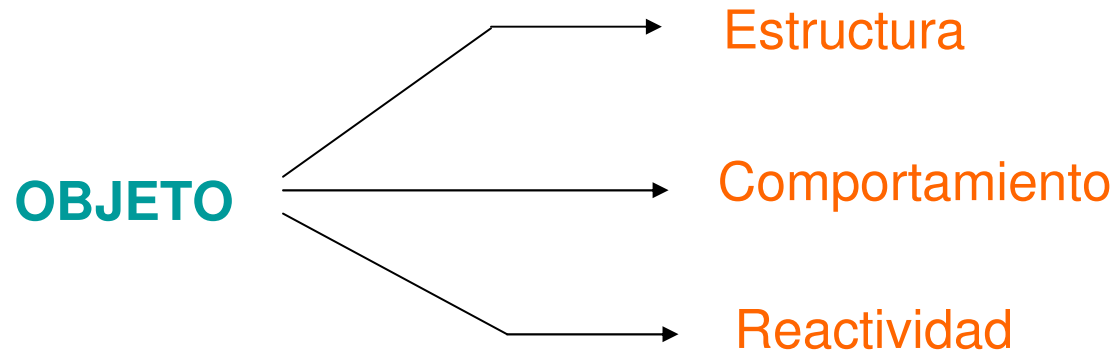
TBox:

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

ABox:

```
reactiveBehavior(has, scientist_review, active).  
reactiveBehavior(received, review, Literal).  
reactivity(scientist_review).
```

Propuesta

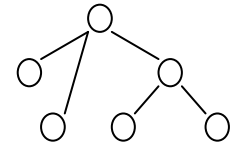


Con los *eventos* como *conceptos dentro de la ontología* se hace factible modelar la *semántica del procesamiento reactivo de los datos dentro del esquema*.

Ventajas:

- El *concepto evento* permite establecer nuevas *relaciones de subsumisión* para crear *jerarquías de generalización entre eventos*, favoreciendo una mejor descripción del dominio de interés.
- Se puede *vincular los eventos a propiedades* específicas cuyo comportamiento es reactivo.
- 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*.
- Mediante el poder de *inferencia* de *una máquina razonadora* es posible *derivar nuevo conocimiento reactivo* implícito en la ontología activa, sin tener que tratar con el problema de la *terminación inherente al paradigma ECA*.

Avances de la Tesis



Formalismo Propuesto

Una **ontología** Oa es una 6-tupla $Oa = \langle F, I, Pe, Pa, E, fr \rangle$ donde:

F : es el conjunto de los hechos extensionales que expresan información sobre las características estáticas y las características activas del dominio de la ontología y de sus individuos.

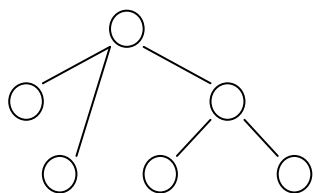
I : es el conjunto de axiomas que permiten inferir nuevos hechos de la ontología con información estática y con información activa concernientes al dominio y a los individuos.

Pe : es el conjunto de hechos referidos a las propiedades estáticas.

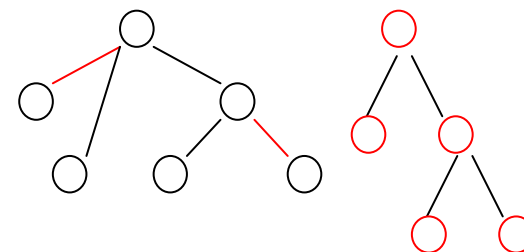
Pa : es el conjunto de hechos referidos a las propiedades activas.

E : es el conjunto de hechos referidos a los eventos de la ontología.

fr : es una función que define el comportamiento reactivo de Oa donde $Ind \times Pa \times E \rightarrow Ind$, con $Ind \subseteq F$.



*isEvent, isSubEventOf,
precedes, ...*



OWL

+

Constructores Activos

=

activeOWL

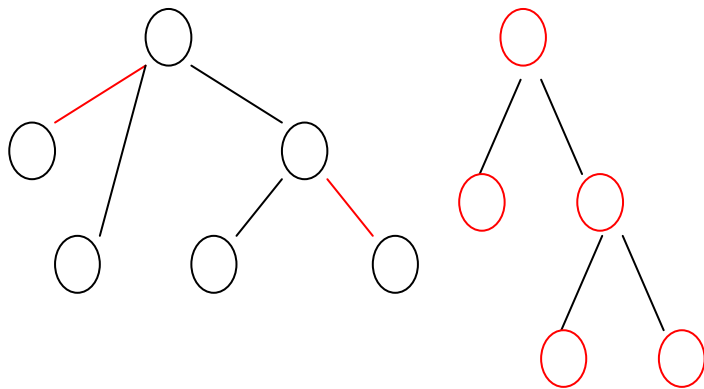
Sintaxis Abstracta	Descripción
<code>isEvent(e)</code>	El evento <i>e</i> .
<code>isSubEventOf(e1 e2)</code>	<i>e1</i> es sub evento de <i>e2</i>
<code>precedes(rp v1 v2)</code>	El valor <i>v1</i> precede al valor <i>v2</i> para la propiedad <i>rp</i>
<code>activeOProperty(rp type rd rr)</code>	La propiedad <i>rp</i> es del tipo <i>type</i> con dominio la clase <i>rd</i> y rango la clase <i>rr</i> .
<code>activeDProperty(rp type rd rr)</code>	La propiedad <i>rp</i> es del tipo <i>type</i> con dominio la clase <i>rd</i> y rango el valor <i>rr</i> .
<code>reactiveBehavior(rp isEvent(e1 be1) value(v1) isEvent(e2 be 2)value(v2) ... isEvent(ek be k)value(vk))</code>	La propiedad <i>rp</i> toma el valor <i>vi</i> cuando ocurre el evento <i>ei</i> y se cumple <i>bei</i> , para todo $i = 1..k$.
<code>ExclusiveIndividual(i rp value (v))</code>	El individuo <i>i</i> tiene el valor <i>v</i> en la propiedad <i>rp</i> como único valor.
<code>simultaneousIndividual(i rp value (v))</code>	El individuo <i>i</i> tiene el valor <i>v</i> en la propiedad <i>rp</i> simultáneamente con otros posibles valores para <i>rp</i> .
<code>orderedIndividual(i rp value (v,t_i))</code>	El individuo <i>i</i> tiene el valor <i>v</i> en la propiedad <i>rp</i> en el momento <i>ti</i> .

Base de Conocimiento Ontológico

Dado que la base de conocimiento de toda ontología comprende un componente que representa conocimiento extensional y otro componente que representa conocimiento intensional



Es posible modelar una ontología como una base de datos deductiva



Formalismo Propuesto



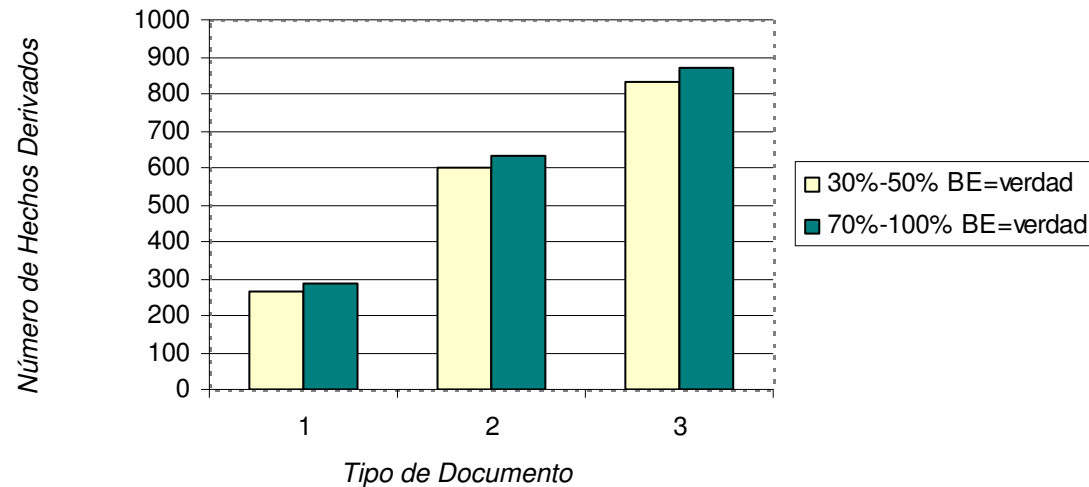
Sintaxis Abstracta de aOWL	Predicados Extensionales AEOB
isEvent(e)	isEventOf(E).
isSubEventOf(e1 e2)	isSubEventOf(E1,E2).
precedes(ap vr1 vr2)	precedes(AP,VR1,VR2)
activeOProperty(ap type(rd) range(rr))	activeOProperty(AP,TYPE,RD,RR)
activeDProperty(ap type(rd) range(rr))	activeDProperty(AP,TYPE,RD,RR)
reactiveBehavior(ap isEvent(e,bc) value v)	reactiveBehavior(AP,E,BC,V).
exclusiveIndividual(i value (ap v)	exclusiveIndividual(I,AP,V).
simultaneousIndividual(i value (ap v)	simultaneousIndividual(I,AP,V).
orderedIndividual(i value(ap (v,ti))	orderedIndividual (I,AP,V,Ti).
Axiomas de aOWL	Predicados Intensionales AIOB
Si subEvent(E1,E2) y subEvent(E2,E3) entonces subEvent(E1,E3).	areSubEvents(E1,E2) :-isSubEventOf(E1,E2). areSubEvents(E1,E2) :-isSubEventOf(E1,E3), areSubEvents(E3,E2).
Si subEventOf(E1,E2) y subEventOf(E2,E3) entonces subEventOf(E1,E3) y E2 y E3 son supraeventos de E1.	areSupEvents(E1,E2):- isSubEventOf(E2,E1). areSupEvents(E1,E2):- isSubEventOf(E,E1), areSupEvents(E,E2).

Resultados preliminares:

Se llevaron a cabo experimentos con ontologías activas sintetizadas.

1. Cómo impacta el tamaño de la jerarquía de eventos.
2. Cómo impacta el número de expresiones booleanas que son verdad.

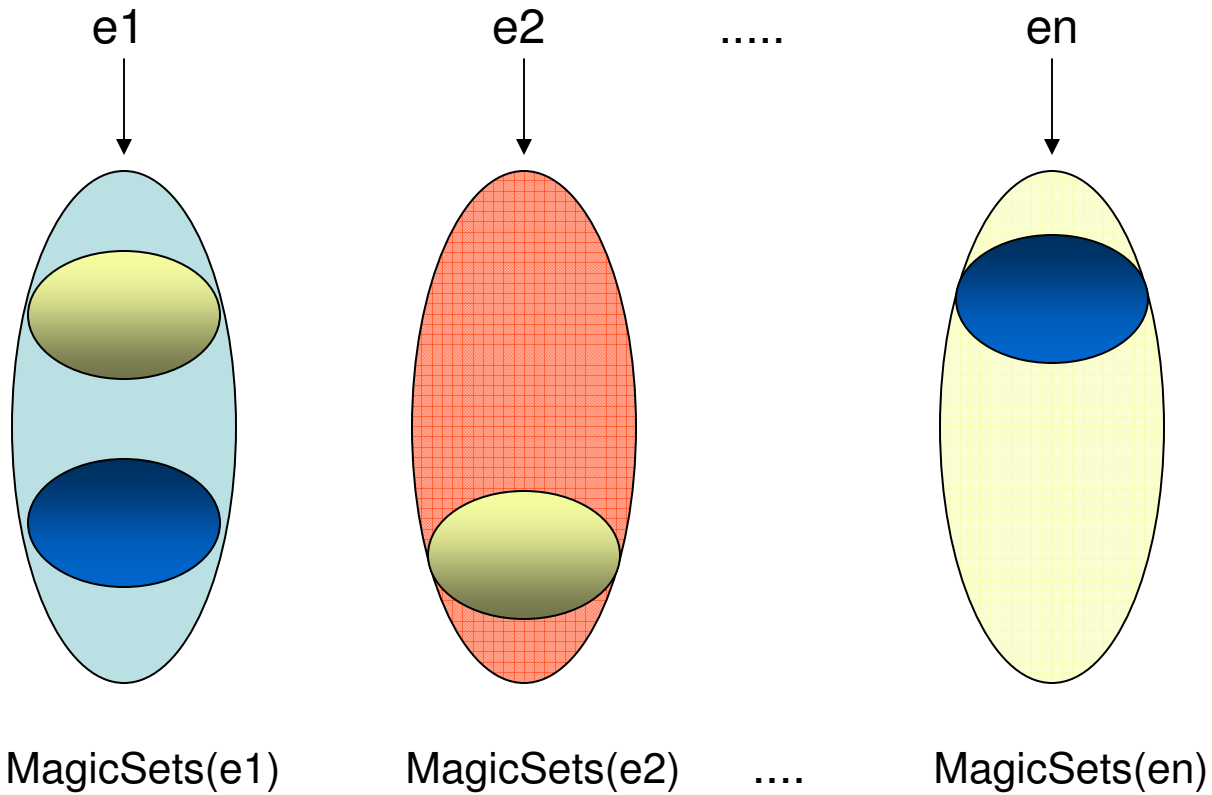
$$N_{MM}(D)$$



Ontologías reales:

Ontología de Lehigh University Benchmark (LUBM)

Ontología del Proyecto Manta con PDVSA-Gas



Propósito:

Extender la noción de Magic Sets para utilizarla como estrategia de optimización a fin de identificar el orden en que deben ser procesados los eventos para administrar eficientemente el comportamiento reactivo de los datos.

Con Magic Sets se 'empujan' las instanciaciones en la consulta a un nuevo programa, particular para esa consulta, donde el número de inferencias repetidas se minimiza.

¿Alguna pregunta? ...