

Extensión del marco de referencia Ontoconcept con el formato de intercambio de reglas RIF

Proyecto de Grado para optar al título de
Magister en Ingeniería de Sistemas y Computación

Lina Marcela Rodas Pérez
Junio 2018

Director del trabajo de grado: Msc. César Augusto Jaramillo Acevedo

Universidad Tecnológica de Pereira
Facultad de Ingenierías: Eléctrica, electrónica, física y
Ciencias de la computación
Maestría En Ingeniería de Sistemas y Computación

Abstract

This project defines a way to include rule management through RIF-BLD in the declarative language proposed in the "Framework for management of ontological changes, based on conceptual models". A comparative study of the expressions of RIF (Rule Interchange Format published by W3C) is made, to the expressive capacity of the Ontology Tree-graph (Grafárbol), detecting advantages. The state of the art in rules with RIF, and ontologies, is reviewed. An extension of Ontoconcept is proposed through new declarative operators. Finally, the Generic Change Ontology, is also extended by adding rules.

Keywords: RIF, OWL, Ontologies, Rule Languages, Semantic Web, Knowledge Representation.

Tabla de Contenidos

Capítulo 1.....	1
Introducción	1
Capítulo 2.....	3
Definición del problema y justificación.....	3
Definición del problema.....	3
Justificación.....	5
Capítulo 3.....	10
Objetivos.....	10
Objetivo General	10
Objetivos Específicos.....	10
Capítulo 4.....	11
Marco teórico y estado del arte.....	11
2013: Segunda versión de RIF	14
Uso de RIF en otros campos investigativos	20
Marco de referencia Ontoconcept	24
Sistemas de reglas	25
Marco conceptual	27
Capítulo 5.....	40

Desarrollo de la propuesta	40
5.1. Análisis de la capacidad expresiva de RIF.....	40
Generalidades	40
Sintaxis de RIF-BLD.....	45
Capacidad expresiva de RIF-BLD	47
5.2. Semántica de RIF-BLD.....	50
Semántica del dialecto.....	50
Axiomas de OWL.....	64
Comparativa con OWL	65
Otros aspectos de comparación.....	67
5.3. Extensión del grafábol ontológico	72
Reglas y Ontologías	72
Definiciones	73
De RIF-BLD al marco de referencia Ontoconcept.....	76
Operadores propuestos	78
Resultados	83
5.4. Extensión de la Ontología de Cambio Genérico	86
Capítulo 6.....	91
Conclusiones.....	91
Recomendaciones y trabajo futuro.....	92

ANEXOS	94
Glosario y acrónimos	95
Referencias.....	97

Lista de tablas

Tabla 1. Expresiones en RIF-BLD en su sintaxis de presentación	48
Tabla 2. Comparativa RIF-BLD y OWL en sentencias	66
Tabla 3. Comparativa de RIF-BLD y OWL en otros aspectos	67
Tabla 4. Operadores conceptuales sobre entidades.....	77
Tabla 5. Operadores sobre aserciones o axiomas, basada en (Chavarro, 2010)	77
Tabla 6. Operaciones sobre metadatos y operaciones de anotación	77
Tabla 7. Operaciones que involucran múltiples ontologías	78
Tabla 8. Operaciones propuestas que involucran reglas	79
Tabla 9. Operadores sobre axiomas (Relation). Basada en (Chavarro, 2010).....	81

Lista de figuras

Figura 1. Búsqueda realizada en Google en marzo de 2017	4
Figura 2: Arquitectura marco de referencia para la gestión conceptual del cambio ontológico, basado en Chavarro (2010).	26
Figura 3. Arquitectura de la web semántica. Basado en Herman (2007)	36
Figura 4: Jerarquía de las Relaciones. Basado en Chavarro (2010)	37
Figura 5. Clases principales de la Ontología de Cambio Genérico. Basado en Chavarro (2010). ..	39
Figura 6. Familia de dialectos RIF. Basado en Paschke, Morgenstern, Hirtle, Ginsberg, Patranjan, y McCabe (2013)	41
Figura 7. Ejemplo de sintaxis RIF-BLD. Basado en Lera (2014).	46
Figura 8. Rule implication en RIF-BLD. Basada en (Lera, 2014).	71
Figura 9. Ontología de Cambio Genérico, clases principales. Basado en (Palma, 2009).	87
Figura 10. Extensión de la ontología de cambio genérico al lenguaje declarativo. Basado en Chavarro (2010)	88
Figura 11. Extensión de la Ontología de Cambio Genérico al lenguaje declarativo con sentencias RIF-BLD.	90

Capítulo 1

Introducción

La presente investigación tiene como tema principal el estudio comparativo de las expresiones de RIF (formato de intercambio de reglas publicado por la W3C) con la capacidad expresiva del grafárbol ontológico, modelo conceptual propuesto dentro del framework Ontoconcept, que es una herramienta para la gestión del cambio ontológico, planteado en la tesis doctoral del ingeniero Julio César Chavarro Porras, bajo el título “Marco de referencia para la gestión del cambio en ontologías, basados en modelos conceptuales”.

La motivación para la presente propuesta se fundamenta en la necesidad de establecer modelos conceptuales para la gestión de ontologías, continuando en una pequeña parte con el proceso de desarrollo de un sistema de administración de ontologías, consulta y razonamiento sobre diferentes ambientes, que es la finalidad a largo plazo de Ontoconcept.

Este trabajo se encuentra vinculado a la línea de profundización de la maestría en ingeniería de sistemas y computación titulada Inteligencia Artificial, asociado al grupo de investigación GIA, grupo en el cual se vienen desarrollando diferentes proyectos de extensión del framework, así como generación de nuevas versiones.

La propuesta se desarrolla en cinco etapas: primero se revisa el estado del arte en el ámbito de ontologías y reglas con RIF, haciendo énfasis en el intercambio de reglas, se hace un análisis de los elementos de la sintaxis de RIF para identificar sus ventajas, se describe una comparativa de

éste con OWL, se establece la extensión del grafárbol propuesta, y por último también se extiende la Ontología de Cambio Genérico.

Capítulo 2

Definición del problema y justificación

Definición del problema

La Web está basada en esquemas de navegación que inicialmente no tenían un orden estructurado y que cada día están generando una enorme cantidad de información que requiere otros mecanismos de procesamiento, clasificación, búsqueda y permanente mantenimiento (Kumar y Kamath, 2015).

Incluso los buscadores web todavía presentan información poco relevante o confusa para el usuario. Por ejemplo, en la figura 1, la búsqueda “Resoluciones ministeriales que salieron hoy” realizada en el motor Google en marzo de 2017, no muestra resultados relevantes para el usuario en ninguna de las primeras cinco referencias, o al menos resultados para Colombia.

Dentro de las propuestas de solución para la recuperación de información se encuentran los agentes inteligentes, el uso de RDF para modelar datos, y las ontologías. La Web Semántica es un término acuñado desde 2001 que encierra estos términos, permitiendo extender la web con tecnologías que proponen un tratamiento automático de la información (Álvarez, 2012).

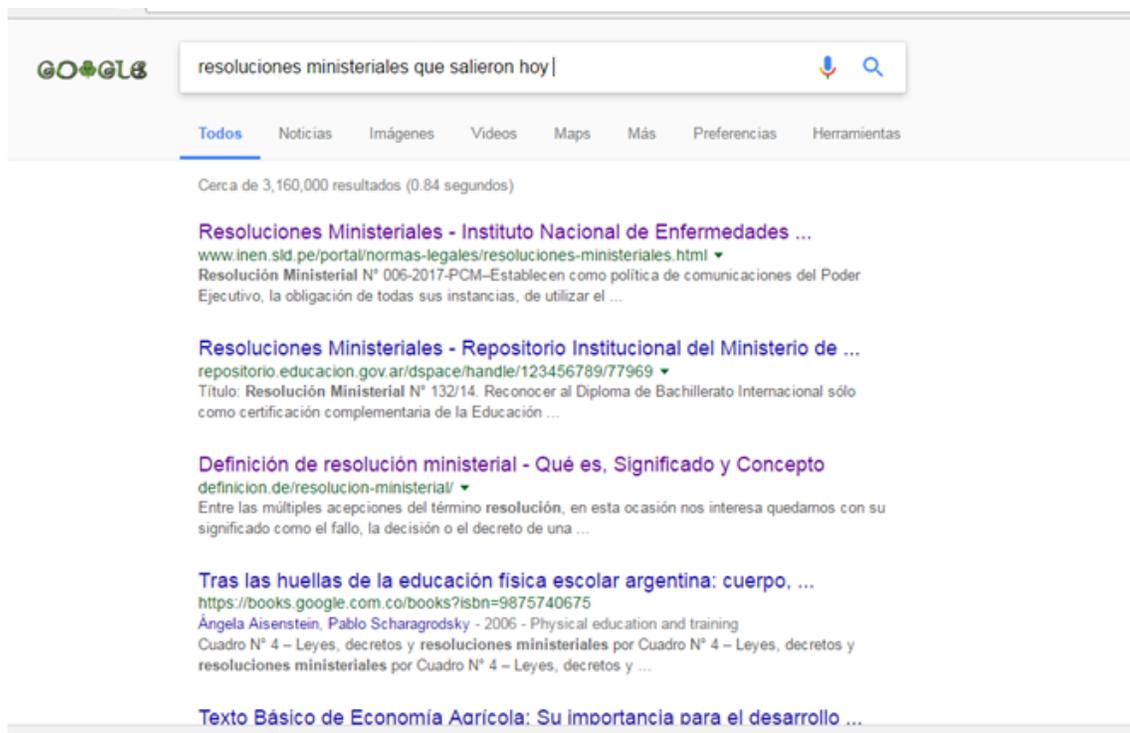


Figura 1. Búsqueda realizada en Google en marzo de 2017

En el campo de las ontologías, se logró la descripción de objetos y de relaciones entre objetos en la web, con el uso de lenguajes enriquecidos tales como: RDFs, OWL (Lite, DL, and Full) y recientemente, RIF. El formato de intercambio de reglas (Rule Interchange Format -RIF) surge como una norma que facilita la integración y síntesis de conjuntos de reglas. Comprende un conjunto de dialectos interconectados que representan lenguajes de reglas con varias características (Morgenstern y Welty, 2013).

Las ontologías son entendidas como especificaciones de una conceptualización de un dominio, generalmente en términos de conceptos, atributos, relaciones y axiomas (Gruber, 1993), las cuales cambian durante su ciclo de vida y se han planteado al menos tres perspectivas para la gestión de dicho cambio: Gestión de versiones, Evolución y Gestión basada en modelos conceptuales.

La propuesta basada en modelos conceptuales de Chavarro (2010), construye un modelo para el ciclo de vida ontológico, en el cual las operaciones de cambio que afectan una ontología son representadas conceptualmente, de manera independiente del lenguaje de implementación. En dicho modelo, se define un conjunto de operadores formales que permite el desarrollo de tareas que involucran múltiples ontologías, como son: mezcla, integración y transformación de ontologías que corresponden a dominios similares, así como una estructura que combina árboles y grafos etiquetados, denominada Grafárbol, para representar tanto la estructura de una ontología como las operaciones que le son aplicables en el nivel conceptual, denotando así la expresividad de las reglas propuestas en los lenguajes.

El problema a resolver es añadir una extensión al lenguaje declarativo, implementación representada en el grafárbol, con las capacidades expresivas de RIF, ya que dicho lenguaje contiene operadores para gestionar el ciclo de vida de las ontologías, pero carece de operadores para gestión de reglas.

El problema se resuelve a través de la definición de la equivalencia semántica entre expresiones RIF y el grafárbol, basados en el marco de referencia propuesto por Chavarro (2010) para añadir características de reglas.

Justificación

La propuesta planteada se enmarca en el trabajo investigativo de Chavarro (2010), se presenta el framework Ontoconcept como un modelo que permite editar, mantener e integrar ontologías, de manera independiente del lenguaje de implementación de las mismas.

Con el framework, se logra modelar conceptualmente la gestión del cambio ontológico, entendida como el proceso que permite representar, implementar, mantener y propagar la actualización de una ontología, en contextos abiertos y distribuidos. Este proceso incluye la creación de la ontología y las tareas que involucran múltiples ontologías que corresponden a dominios similares, como son la reutilización, mezcla, alineación, e integración de ontologías.

Ontoconcept presenta las siguientes características:

- La definición de un modelo conceptual para la gestión del cambio ontológico: Se compone de una representación de las ontologías a través de grafárboles y del desarrollo de un conjunto de operadores para el trabajo con dichos grafárboles, conformando un lenguaje declarativo. El grafárbol es un concepto que permite representar una ontología diferenciando entre tipos de nodos y tipos de relación. Las relaciones utilizadas en una ontología se pueden describir por medio de una jerarquía de relaciones.
- Una extensión de ODM, Ontology Definition Metamodel (Object Management Group, 2014), en la cual se incluyen operaciones de cambio ontológico.
- La descripción de una arquitectura tecnológica que soporte el lenguaje declarativo propuesto.

Extender el grafárbol ontológico, con características de gestión de reglas de acuerdo con el formato RIF, permitiría continuar con su propósito de implementación a largo plazo de un sistema de administración de ontologías, consulta y razonamiento sobre diferentes repositorios, ya que representaría una actualización para Ontoconcept, el cual es objeto de estudio en la creación de nuevas versiones, dentro del grupo de investigación GIA.

El formato de Intercambio de Reglas, además, está diseñado como un estándar, que permite la interoperabilidad entre lenguajes de reglas en general y sus usos no están limitados a la web, lo que lo convierte en una metodología pertinente para su utilización en reglas de negocio, por ejemplo, el acceso a las reglas comerciales de los socios de una cadena de suministro (Paschke, Morgenstern, Hirtle, Ginsberg, Patranjan, y McCabe, 2013).

El diseño de RIF refleja la lógica de identificar tipos específicos de reglas dentro de sistemas de reglas existentes, llamados Dialectos RIF, que pueden ser traducidos a otros sistemas de reglas sin cambiar su significado (Boley y Kifer, 2013).

La ventaja de implementar RIF consiste en el tratamiento de metadatos que pueden provenir de diferentes sitios web, y estar representados en diferentes lenguajes. Como desarrollo paralelo a OWL, pueden ser inter compatibles, e incluso existen subconjuntos de OWL que se pueden implementar como subconjuntos de RIF (Boley y Kifer, 2013).

Así que dichas tecnologías de web semántica se hacen muy importantes en el presente y futuro cercano, para agilizar en la Web la tarea de búsqueda de información y el procesamiento a través de máquinas.

El desarrollo de esta propuesta permite a la autora aplicar los conocimientos teóricos aprendidos en el proceso de la presente maestría, dentro del ámbito de la representación del conocimiento orientada a tecnologías de la web semántica.

Delimitación. RIF es un conjunto de dialectos (lenguajes de reglas rigurosamente definidos), creado para facilitar el intercambio de reglas y su uso compartido (Morgenstern y Welty, 2013). El intercambio de reglas permite trabajar con lenguajes de reglas provenientes de diferentes paradigmas (primer orden puro, bases de datos con programación lógica/deductivas, reglas de producción, reglas reactivas, entre otros). Lo anterior en vista que dichos lenguajes pueden tener características diferentes: sintaxis, intereses comerciales, y distintas preferencias de sus usuarios.

RIF se compone de varios dialectos interrelacionados, cuidadosamente escogidos, que sirven al propósito de compartir e intercambiar reglas sobre la web; tales dialectos corresponden a las muchas diferentes semánticas de los lenguajes: primer orden clásico, semántica de modelo estable para la negación, semántica bien fundada para la negación, entre otros. Con RIF existe una mejor oportunidad de traer los lenguajes de reglas a los desarrollos actuales en Internet y se puede hacer la programación web mucho más intuitiva.

A continuación se describen brevemente los principales dialectos que componen RIF:

RIF-core: el lenguaje RIF básico. Todos los dialectos de RIF son una extensión de RIF-Core + DTB (Morgenstern y Welty, 2013).

DTB: un conjunto de tipos de datos, funciones incorporadas y predicados que pueden ser usados cuando escribimos reglas.

RIF-BLD: Basic Logic Dialect.

RIF-PRD: Production Rule dialect.

El trabajo propuesto en el presente documento utilizará el dialecto RIF Basic Logic Dialect (RIF-BLD) de acuerdo con la versión del año 2013 del estándar RIF recomendado por la W3C (Morgenstern y Welty, 2013).

De igual manera, para su verificación se utilizará la semántica de cada lenguaje, representada en lógica de descripción, lo que indica que no se implementará en ninguna plataforma tecnológica, sino que las comprobaciones se realizarán convirtiendo documentos OWL a RIF y viceversa.

No se implementará el dialecto RIF Production Rule Dialect (PRD) señalado dentro del mismo conjunto de especificaciones publicadas en febrero de 2013.

Capítulo 3

Objetivos

Objetivo General

Extender el grafárbol ontológico que da soporte a Ontoconcept, con las capacidades expresivas del Formato de Intercambio de Reglas (RIF), restringido al dialecto RIF Basic Logic Dialect (RIF-BLD).

Objetivos Específicos

- Identificar los elementos que caracterizan el formato de intercambio de reglas (RIF) y que determinan su capacidad expresiva.
- Establecer la semántica de cada expresión en RIF, para establecer su correspondencia con los lenguajes OWL y RDF.
- Proponer una extensión del grafárbol ontológico estableciendo la conversión de las expresiones en RIF a operaciones sobre el grafárbol.
- Extender la Ontología de Cambio Genérico con las operaciones de cambio que pueden provenir del lenguaje RIF en el dialecto BLD.

Capítulo 4

Marco teórico y estado del arte

El problema de la implementación del Formato de Intercambio de Reglas RIF en todas sus extensiones, es un campo reciente en la ingeniería web. El estándar más reciente de la W3C es del año 2013 (Kifer y Boley, 2013); La mayoría de los trabajos desarrollados que involucran los dialectos RIF se basan en la primera versión del mismo (2010), e incluso, las investigaciones sobre lenguajes de reglas usados en la web para motores de búsqueda semánticos, proceden de fechas anteriores (Fillottrani, 2005), (Veiga, 2011). Además, no hay desarrollos en Colombia en el tema especificado.

Como se evidencia en el trabajo de Fillottrani (2005), el autor pretendía realizar una investigación acerca del intercambio de reglas, anterior a la publicación de RIF como estándar. Sin embargo, no existe un registro que tal desarrollo se haya culminado o se hayan hecho divulgaciones posteriores, debido a que el grupo de estudio de la World Wide Web Consortium (W3C) acerca del formato de intercambio de reglas tenía objetivos similares.

Para el primer estándar, destaca la tesis de maestría de Frédéric Veiga (2011), en la Universidad Nueva de Lisboa, con una implementación del *RIF Production Rule Dialect* (Sainte Marie y Hallmark, 2013), usando Jess, pero presentando un primer desarrollo aplicativo de RIF-PRD en forma declarativa. En dicha disertación, el autor sostenía que no existían implementaciones completas de RIF-PRD.

En (Morán, 2011), se presenta un análisis de RIF como el único lenguaje capaz de soportar las características de alto nivel para las reglas de escalabilidad, aunque se compara con otras

diferentes alternativas. Se trabaja con un subconjunto de RID-PRD para cumplir con los requisitos de la investigación, demostrando capacidades de interoperabilidad a través de un mapeo entre sentencias de dos lenguajes de reglas de bajo nivel; para ello, se tuvo que construir un trasladador de PRD a cada uno de los dos lenguajes; dicho componente traductor se alimenta con el dialecto RIF y las correspondientes ontologías.

En (Carral Martínez y Hitzler, 2012) se muestra una investigación similar al presente trabajo, en el cual se desarrolla un modelo para extender OWL a través de las reglas de lógica descriptiva (la lógica que apuntala a OWL, DL) proponiendo un enfoque de traducción de reglas en OWL extendido por conjunción de funciones. Se desarrolla un ejemplo con un fragmento de Datalog, expresado en lógica de descripción y mostrando reglas más complejas dentro de la notación DL. Sin embargo, el autor no aplica RIF directamente, sino que extiende reglas y agrega la conjunción. También debemos anotar que en el presente objeto de investigación se partirá del grafárbol, concepción de la propuesta basada en modelos conceptuales de Chavarro (2010), es decir que no se iniciará como se menciona en este artículo.

En el trabajo de investigación de Wang, Chen, Cao y Meng (2013), se propone f-RIF, una combinación de RIF y conjuntos difusos, la cual sirve para representar e intercambiar reglas difusas, teniendo en cuenta que RIF por sí solo es apropiado para intercambio de reglas no difusas. Aquí se explica la sintaxis de f-RIF, su perfil UML y su semántica. La intencionalidad de estos autores es que f-RIF se pueda aplicar sobre lenguajes de reglas difusas heterogéneos.

De la misma manera, y continuando con f-RIF, en Wang, Meng, Chen, y Wang (2013) se define el mapeo necesario para compartir reglas entre f-R2ML y f-RIF; f-R2ML es un lenguaje de marcado de reglas difusas importante en la web semántica, pero estos lenguajes presentan sintaxis heterogéneas, por lo cual en dicha propuesta se describe el proceso para hacer

transformaciones entre sus sintaxis concretas y abstractas, con el fin de incluir f-R2ML en el framework de intercambio f-RIF. También se analiza la información perdida en el proceso de transformación y se proponen medidas de solución para reducir la posible pérdida de información.

En Pomarolli, Anderlik, y Küng (2012) se especifica una propuesta de R2RIF (Rule Integrator to RIF), un plugin de integración de reglas para Protègè OWL, con el objetivo de mezclar ontologías y reglas para construir modelos conceptuales acerca de cualquier dominio. De acuerdo con las tareas de transformación y conversión, R2RIF ofrece algoritmos genéricos y flujos de trabajo para ser adaptables a diferentes lenguajes ontológicos y de reglas. Por lo tanto, R2RIF se presenta como un complemento para OWL, destinado a importar, transformar y convertir lenguajes basados en reglas de lógica de Horn a RIF, para posteriores usos.

El plugin propuesto pretende integrarse en la arquitectura de dos componentes de lenguajes de reglas con el objetivo de ofrecer en la ingeniería de ontologías, métodos para mejorar la semántica actual con reglas, para modelar problemas que no pueden ser resueltos con ontologías, por ejemplo, los modelos multinivel.

Hasta aquí se presentan los resultados investigativos acerca del formato de intercambio de reglas, de acuerdo con el estándar del año 2010.

2013: Segunda versión de RIF

Posterior a la publicación de las recomendaciones de la W3C en cuanto a los dialectos básicos de RIF, se resalta el trabajo de Rattanasawad, Runapongsa Saikaew, Buranarach, y Supnithi (2013), donde se realiza una comparación de la expresividad de lenguajes de reglas (Incluido RIF 2013), las operaciones soportadas, una comparación de los motores de inferencia: las estrategias de razonamiento y algoritmos usados, los lenguajes de programación y de reglas soportados, entre otras características (Rattanasawad, *et al.*, 2013).

Los autores parten de FOLRuleML, SWRL, Notation3, Jena Rule Format y RIF como lenguajes de reglas, y revisan el motor Jena, EYE, OWLIM-Lite, BaseVISor y FuXi como razonadores basados en reglas dirigidos al procesamiento de datos en la web semántica, estableciendo tablas comparativas. El propósito que se identifica aquí es establecer una ayuda para desarrolladores, de manera que seleccionen el lenguaje de reglas adecuado que cumpla con los requisitos para sus aplicaciones.

De manera similar, en el presente proyecto se pretende comparar RIF con las capacidades expresivas dadas en el marco de referencia Ontoconcept, es decir qué se utilizarán también criterios de expresiones y operaciones soportadas, pero con el fin de extender las funciones que presente el modelo conceptual base.

En Kim (2015), se aborda el problema del conflicto de autorización que surge con la inferencia en RDF. Cuando se especifica una autorización de acceso, puede estar comprometida con otras autorizaciones de acceso que tienen señal de seguridad opuesta como resultado de la inferencia de RDF. Al combinar RIF con datos RDF, por lo tanto, como en la inferencia de RDF, se puede causar un conflicto de autorización por la inferencia de RIF. Además, este conflicto de

autorización puede surgir como resultado de la interacción de la inferencia de RIF y la inferencia de RDF en lugar de solamente la inferencia de RIF. Así que el autor sugiere un algoritmo de detección de conflicto de autorización que resulta muy eficiente porque se basa en el etiquetado de gráficos y demuestra con experimentos que el rendimiento es bueno para datos grandes.

En Colombo, Pugliese, Klein, y Lützemnberg (2014), se desarrolla un trabajo investigativo para utilizar RIF para representar, en un esquema XML, la unión y traducción entre dos lenguajes de modelado y formalización del conocimiento: un patrón basado en el diagrama de clases UML, y otro basado en el diagrama IDEF0, Integration Definition for Function Modeling (Knowledge Based Systems, 2017). Lo anterior con el fin de reusar piezas de ingeniería del conocimiento que ya han sido representadas en otras herramientas de software.

Los autores hacen una comparación de diferentes sistemas de ingeniería basada en conocimiento (KBE), en cuanto a sus capacidades y limitaciones. Por último utilizan RIF para “capturar conocimiento en un formato neutral, para que pueda ser utilizado en diferentes ambientes. El conocimiento representado incluye bloques de codificación para arquitectura de productos, procesos de diseño, información geométrica y conexiones a herramientas externas, como sistemas CAD o PDM” (Colombo, *et al.*, 2014).

En Maeda, Ohta, y Kuwabara (2014), no se aplica RIF propiamente pero se propone un método que implementa un sistema de reglas basado en el framework MapReduce, que es un framework para escribir fácilmente aplicaciones que procesan grandes cantidades de datos (conjuntos de datos de varios terabytes) en paralelo en clústeres grandes (miles de nodos) de hardware básico de una manera confiable y tolerante a fallas (Apache Software Foundation, 2018).

En Maeda, *et al.* (2014) se construye un sistema de reglas simple usando Hadoop, implementación de fuente abierta de MapReduce, y se comparan varios métodos para ejecutar el sistema de reglas.

Kalinichenko, Stupnikov, Vovchenko, y Kovalev (2014) exponen en su artículo denominado “Rule-Based Multi-dialect Infrastructure for Conceptual Problem Solving over Heterogeneous Distributed Information Resources”, una propuesta para aplicar una combinación de lenguajes basados en reglas, diferentes semánticamente, en programación conceptual interoperable sobre varios sistemas basados en reglas.

El esquema conceptual de un problema (las clases) se define en el marco de un dominio y consta de un conjunto de documentos RIF (el documento es una unidad de especificación de RIF). Cada documento contiene grupos de reglas. La conceptualización del dominio se realiza utilizando ontologías OWL 2 que contienen entidades del dominio y sus relaciones. Las ontologías constituyen la especificación conceptual del dominio. Al final de la propuesta, se describe la aplicación de esta infraestructura en la integración de bases de datos heterogéneas y se detalla un ejemplo de aplicación para resolver problemas combinatoriales complejos.

En Chen, Xu, Ni, Cao, y Zhang (2014), destaca el trabajo realizado sobre bases de datos relacionales (RDBs), desarrollando un prototipo para mapeo RDB a RDF (comúnmente abreviado como RDB2RDF) el cual toma los aspectos de expresión de reglas de RIF. Este problema de mapeo ha sido tratado desde distintas perspectivas anteriormente y es la clave para la construcción de la Web de Datos.

En 2009, el informe del grupo gestor RDB2RDF de la W3C, sugirió fuertemente que el lenguaje de mapeo RDB2RDF fuera expresado en reglas como lo define el grupo de trabajo de RIF; en este sentido, los autores estudiaron el problema del mapeo de RDB2RDF basado en RIF,

proponiendo un enfoque de descripción de mapeo basado en RIF-PRD, y orientado a bases de datos semánticas. El trabajo incluye la definición de un conjunto de reglas de mapeo genéricas RIF-PRD para RDB2RDF, desarrollando un prototipo de motor de mapeo llamado RIFD2RME (por motor de mapeo RDB2RDF basado en RIF), y describiendo la arquitectura y componentes principales de dicho prototipo.

Continuando con aplicaciones sobre bases de datos, en Ouaret, Boussaid, y Chalal (2014) se propone un enfoque automático para generar el esquema de un almacén de datos (Data Warehouse) XML a partir de XML Schema, describiendo fuentes XML utilizando UML, considerándolo muy compatible y familiar para grupos grandes de usuarios.

Se diseña el esquema lógico para una plataforma de datos basada en el concepto anterior, ya que los almacenes de datos proporcionan una herramienta eficiente y un repositorio consolidado con una vista multidimensional de los datos. Varios estudios anteriores han abordado el problema para diseñar un esquema conceptual y lógico de XML Data Warehouses. Sin embargo, estos enfoques se centran solo en la identificación de conceptos multidimensionales de forma semiautomática.

Los autores pretendieron no solo automatizar un diseño de almacenamiento de datos XML, sino también proporcionar un proceso de simplificación y un conjunto de reglas que aplique transformaciones sucesivas para crear el esquema en estrella (Ouaret, Boussaid, y Chalal, 2014). También se presenta aquí una herramienta gráfica para ayudar a ingenieros a diseñar según el modelado de datos multidimensionales. Sin embargo, en este trabajo no se aplica intercambio de reglas, pero proporciona un ejemplo que utilizaremos más adelante, de cómo transformar la semántica a UML.

En Verborgh, y De Roo (2015) se examina el software EYE, que es una herramienta de razonamiento sobre la web, escrita en Prolog. Los datos vinculados con dicha aplicación representan cada dato como un enlace entre dos cosas; esto permite que los razonadores de software lleguen a conclusiones de una manera humana. Igualmente en este artículo se discute cómo el razonador EYE explota datos enlazados y cómo la industria está empleando EYE. Se examinan otras características de su arquitectura, como la portabilidad, debido a que la interoperabilidad es crucial en la web semántica, lo que significa que los componentes de EYE son intercambiables.

Utiliza la sintaxis Turtle, que fue estandarizada en 2014. Para la comunicación a través de EYE se deben acordar un conjunto predefinido de built-ins con un significado rigurosamente específico. Si los agentes no lo hacen, los resultados y las correspondientes pruebas serán poco confiables. Aunque RIF incluye tal conjunto, la adopción ha sido mínima. EYE y otros razonadores han intentado mejorar en este aspecto.

Aquí se puede concluir que RIF posee muchas ventajas sobre otras herramientas de razonamiento, pero falta difusión y aplicación.

En Boley (2016) se describe el centro de conocimiento-interoperación de RuleML, Lenguaje de marcado de reglas, que define un lenguaje de marcado de reglas compartido y permite reglas de avance -de abajo hacia arriba- y hacia atrás -de arriba hacia abajo- en XML (Boley y Tabet, 2013). Este centro proporciona una representación sintáctica/semántica y una transformación interna/externa del conocimiento formal que se puede recopilar en bases de conocimiento (KB) y transformar en una red como una intranet o Internet, específicamente en la Web.

El sistema de representación permite la configuración de sintaxis *textbook* y sintaxis *Relax NG* enriquecida, así como la asociación de la sintaxis con la semántica. El conjunto de herramientas

de transformación incluye formateadores serializados (normalizadores y compactadores), analizadores sintácticos polarizados y generadores (la herramienta RuleML \leftrightarrow POSL, el generador RuleML \rightarrow PSOA / PS y el analizador PSOA / PS \rightarrow AST), así como importadores y exportadores (el importador de Dexlog a Naf Datalog RuleML y el exportador de lenguajes FOL RuleML a TPTP). También se describe la convergencia de RIF y RuleML facilitada por PSOA RuleML.

En Gu y Sun (2017) se evidencia una metodología más reciente de aplicación de gestión de reglas, esta vez en detección de patrones frecuentes, campo que tiene una base de conocimiento híbrida que consiste en ontologías y lógica de reglas. Sin embargo, la eficiencia de la minería es demasiado baja e insatisfactoria, ya que el proceso de razonamiento necesita grandes cantidades de tiempo de computación. Por ello en este trabajo se desarrolla un algoritmo paralelo y se describe la arquitectura con la que se aborda este problema. Se basa en el framework MapReduce (descrito anteriormente), y describe un método de detección de patrones en Programación Lógica Inductiva (ILP) paralela, usando el algoritmo de expansión de nodos. Se distribuyen los diversos procesos de detección de patrones computacionales a cada nodo de computación paralela en el árbol de patrones frecuentes (FPT) y se obtiene el patrón frecuente. De esta forma también se puede eliminar la redundancia semántica, que se juzga mediante grados de soporte.

En Zambon y Guizzardi (2017), el trabajo titulado “Formal Definition of a General Ontology Pattern Language using a Graph Grammar”, presenta un enfoque basado en modelado conceptual, similar a como se logró en Chavarro (2010), partiendo de OntoUML, el lenguaje de modelado conceptual basado en ontologías, que incorpora una serie de patrones ontológicos que reflejan las micro-teorías que comprenden la UFO (Unified Foundational Ontology). En este

contexto, surge la propuesta de la Gramática de Grafos, donde los autores utilizan transformaciones de grafos para modelar formalmente la construcción de patrones de ontología. Un conjunto de reglas de transformación de grafos se puede ver como una especificación declarativa de cómo la construcción de un modelo de ontología puede evolucionar a partir de un estado inicial, representado por un grafo de host inicial. Esta combinación de un conjunto de reglas más un grafo inicial se llama Gramática De Grafos y el conjunto de grafos (posiblemente infinito) alcanzable del grafo inicial constituye el lenguaje gramatical.

Por tanto, se redefine OntoUML como una gramática de grafos formal y se demuestra cómo los modelos de este lenguaje pueden construirse mediante la aplicación combinada de patrones ontológicos siguiendo una serie de reglas de transformación de grafos. Como resultado, se obtiene una versión de este lenguaje completamente definida como una Gramática de Patrón Ontológico formal. Es decir, se logra una definición formal de OntoUML que es explícita en términos de los patrones ontológicos que incorpora y a la vez es completamente independiente del meta-modelo de UML.

Uso de RIF en otros campos investigativos

En Nikkilä, Nash, Wiebensohn, Seilonen, y Koskinen (2012), se utiliza el intercambio de reglas junto con sistemas de información geográfica (para realizar inferencia espacial, producción asistida por computador de cartografía digital), con el fin de tener interoperabilidad. Se presenta un sistema que combina un conjunto de reglas de RIF con la funcionalidad completa del OGC SFA, Open Geospatial Consortium Simple Feature Access (Open Geospatial

Consortium, 2018) e integrando datos a través de RDF. Los autores sugieren que hay una ventaja con el uso de RIF, debido a que se elimina la necesidad de traductores para las reglas.

En Skvortsov, Vovchenko, Kalinichenko, Kovalev, y Stupnikov (2014), se describe una aplicación de RIF para la especificación de flujos de trabajo (workflows). El desarrollo de flujos de trabajo y la acumulación por parte de las comunidades científicas de métodos abiertos supone su especificación en un entorno abierto y su búsqueda, para utilizarlos en la resolución de problemas.

El modelo de metadatos requerido para la búsqueda semántica incluye la descripción de la estructura del flujo de trabajo, enlazando con conceptos del dominio, calidad y requisitos de procedencia de datos y métodos. En la aplicación descrita los metadatos se definen en términos de ontologías correspondientes.

En Nan, Van Gorp, Korsten, Kaymak, Vdovjak, Lu, y Duan (2015) hay un ejemplo investigativo, aplicado a las listas de chequeo de seguridad clínicas, que aunque no hace uso de sistemas de reglas o de intercambio a través de RIF, es importante mencionarlo debido al aprovechamiento de meta-modelos y el estudio comparativo que hace de lenguajes de modelado.

Las listas de chequeo clínicas son un aspecto de investigación relevante ya que pueden reducir los errores médicos y mejorar la seguridad del paciente. Los sistemas de información para este ámbito deben individualizar las listas de chequeo basadas en la información de la historia clínica del paciente, y también considerar el contexto de los flujos de trabajo (workflows) clínicos.

Desafortunadamente, las definiciones de formularios, las consultas a bases de datos y las definiciones de workflow relacionadas con listas dinámicas a menudo están escritas de forma estricta en el código fuente de los sistemas de soporte. Debido a esto hay un mayor esfuerzo

cognitivo en el proceso de diseño, y se complica el intercambio de definiciones de listas dinámicas, así como la interoperabilidad con otros sistemas de información. Por eso se construyó el meta-modelo DCCSS (Dynamic Clinical Checklist Support System), como una extensión incremental de meta-modelos estándar, que permite la reutilización de editores de modelos genéricos en una nueva configuración.

DCCSS integra BPMN, Business Process Model and Notation (Object Management Group, 2018), con Guideline Interchange Format (GLIF), que representan los mejores lenguajes de desarrollo para el modelado de flujo de trabajo clínico y el modelado de reglas clínicas, respectivamente.

En Gupta, Aggarwal, y Ghose (2016) se especifica un concepto de anotación sobre XML para almacenar facturas comerciales, aunque no es aplicado el formato de intercambio de reglas. Para representar datos de una factura con una estructura irregular u oculta, los datos semiestructurados permiten un formato de descripción "sin esquema" en el que los datos están limitados de manera restringida en comparación con un sistema de bases de datos habitual. De esta manera, los autores utilizan gestión de reglas para hacer la anotación y lograr agregar más estructura a las facturas de negocios a fin de simplificar y estandarizar el almacenamiento y la recuperación de la información comercial. Se usan etiquetas XML para anotar manualmente el contenido de una factura y se definen reglas previamente para capturar información.

En Lee, Wang, Hsiao, y Tsai (2017) se desarrolla una ontología para la evaluación de la calidad de patentes, junto con un agente de evaluación y recomendación de tecnología de patentes basado en un enfoque de *soft-computing*, computación suave, para mejorar la expansión de patentes y la transferencia de tecnología.

En primer lugar, se integraron datos de patentes seleccionados con las características de las organizaciones propietarias (instituciones) para establecer una ontología de patentes común para institutos u organizaciones académicas generales. Se utilizó el índice de evaluación de la Oficina de Patentes de Japón y el cociente de propiedad intelectual para describir la base de conocimiento y la base de reglas del agente de evaluación de la calidad de las patentes mediante el lenguaje de marcado difuso (FML, fuzzy markup language); se combina entonces FML con un algoritmo genético. El agente considera la idoneidad de la tecnología patentada para dicha organización, los resultados del proceso de evaluación integral de calidad de patente y los resultados de la evaluación de los requisitos de patente del demandante.

De acuerdo con los resultados, el agente propuesto es factible para una recomendación de patente. Además, se proyecta en un futuro cercano aplicar agentes inteligentes que ayuden a las organizaciones a recomendar una patente, basados en el modelo de ontología construida, la base de conocimiento y la base de reglas que se encuentran en Internet.

Por lo anterior, la segunda versión del estándar RIF es un campo abierto para su estudio y aplicabilidad, donde se pueden hacer desarrollos utilizando lenguajes y motores de inferencia diferentes, así como una orientación a muchos otros problemas académicos y comerciales.

Además, su abordaje desde el formalismo de los modelos conceptuales, y más específicamente, desde la concepción de grafárbol ontológico como herramienta para el modelado conceptual, permitirá profundizar en este ámbito y aportar en la construcción de sistemas de gestión de ontologías que sean independientes del lenguaje de implementación.

Marco de referencia Ontoconcept

Para tomar como punto de partida el grafárbol ontológico y extenderlo con características del formato de intercambio de reglas RIF, es necesario conocer las nociones teóricas del prototipo desarrollado en (Chavarro, 2010). El marco de referencia Ontoconcept especificado, se divide así:

- La definición de un modelo conceptual para la gestión del cambio ontológico (grafárbol y lenguaje declarativo).
- Una extensión de ODM, en la cual se incluyen operaciones de cambio ontológico.
- La descripción de una arquitectura tecnológica que soporte el lenguaje declarativo propuesto.

En la definición del modelo conceptual para la gestión del cambio ontológico el grafárbol es entendido como una herramienta para el modelado conceptual, con la capacidad para representar modelos de conocimiento que puedan ser expresados en lógica descriptiva, es decir, con la capacidad de representar ontologías. Esto significa un aporte importante en la perspectiva de modelos conceptuales, eje principal para superar las limitaciones de dependencia de los lenguajes de implementación de ontologías.

El problema objeto de estudio de la presente propuesta, se centra en la verificación de la existencia de una equivalencia semántica entre operaciones descritas en RIF, comparando con OWL, y operaciones del lenguaje declarativo paralelas a la representación en el grafárbol.

La extensión de ODM, para incluir operaciones de cambio ontológico, se realiza en Chavarro (2010) para facilitar el proceso de desarrollo de nuevas ontologías, incorporando en las herramientas de modelado ontológico, el proceso de gestión del cambio. Ello se logró a través de

la extensión de la ontología de cambio genérico, modelo propuesto en Palma (2009), lo que permite el uso de UML como herramienta de modelado, en los problemas relacionados con la gestión del cambio ontológico.

La arquitectura tecnológica que soporta el lenguaje declarativo propuesto, se refiere a las especificaciones del framework propuesto para gestión del cambio en ontologías, cuyas características se resumen en la figura 2.

Sistemas de reglas

Una regla es una guía prescrita para la conducta o una regulación para gobernar un comportamiento. En programación lógica hay dos maneras de ver las reglas: una regla es una instrucción en un programa. Si cierta condición se cumple, se lleva a cabo una acción. Tales reglas se refieren como reglas de producción (Morgenstern y Welty 2013).

Alternativamente uno puede pensar en una regla como un indicador de un hecho sobre el mundo. Estas reglas se conocen como reglas declarativas: Si P entonces R. Las reglas declarativas no expresan una acción que se lleva a cabo.

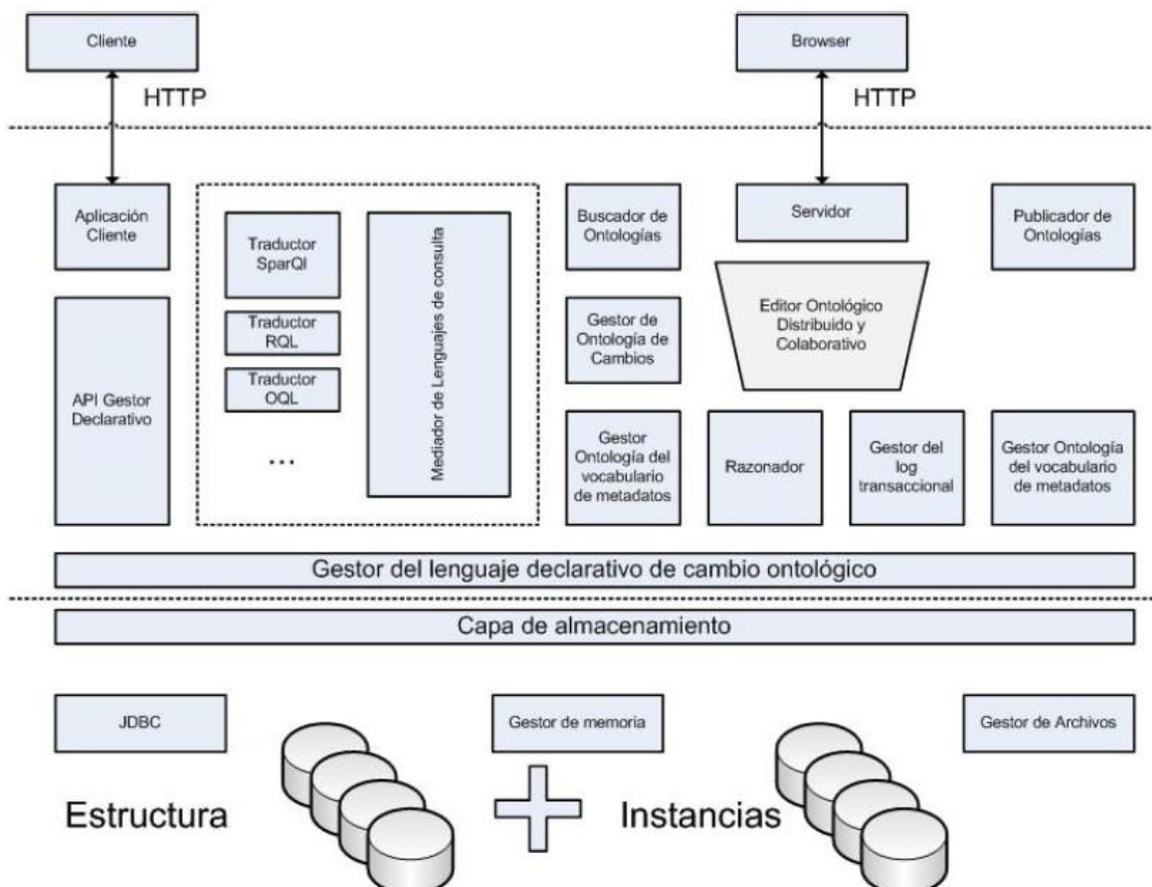


Figura 2: Arquitectura marco de referencia para la gestión conceptual del cambio ontológico, basado en Chavarro (2010).

Las reglas y los hechos deben ser escritos en alguna clase de lenguaje formal para que los programas de computador razonen con ellos y saquen conclusiones. Tales programas a menudo son llamados *motores de reglas* (Morgenstern y Welty 2013).

Lenguajes de reglas existentes. Declarativos:

SILK

OntoBroker

Eye

VampirePrime

N3-Logic

SWRL

De producción:

Jess

Drools

IBM ILog

Oracle Business Rules

Características declarativas y de producción:

Prolog

Todos los lenguajes de reglas de producción tienen un subconjunto básico que es declarativo.

El formato de intercambio de reglas RIF fue creado para trabajar con conjuntos de reglas de diferentes lenguajes.

Marco conceptual

A continuación se refiere una lista de términos y conceptos que permiten sintetizar el contexto tecnológico donde se encuentra el problema objeto de estudio:

Ontología. “Una ontología es una especificación explícita de una conceptualización compartida”, según Thomas Gruber (1993). Una conceptualización es una abstracción, una vista simplificada del mundo que queremos representar.

Una ontología especifica una conceptualización de un dominio, generalmente en términos de conceptos, atributos, relaciones y axiomas. Constituye por tanto una red semántica que puede ser representada por diversos lenguajes (por ejemplo, DAML+OIL, OWL, entre otros.). Las

ontologías, frente a otros sistemas de representación de conocimiento, permiten expresar más contenido semántico y realizar inferencias.

Una ontología no es una base de datos ni un programa (porque tienen sus propios formatos internos), no es una conceptualización (porque no es una especificación, es sólo una vista) ni una tabla de contenidos (aunque una taxonomía sí es una ontología).

Las ontologías son acuerdos, en un contexto social, para cubrir una serie de objetivos. Se crean, entre otras cosas, para:

- Permitir el intercambio de datos entre programas.
- Simplificar la unificación (o traducción) de distintas representaciones.
- Facilitar la comunicación entre personas.

Continuando con Gruber, las ontologías tienen los siguientes componentes que servirán para representar el conocimiento de algún dominio:

- **Conceptos:** son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, entre otros.
- **Relaciones:** representan la interacción y enlace entre los conceptos del dominio. Suelen formar la taxonomía del dominio. Por ejemplo: subclase-de, parte-de, parte-exhaustiva-de, conectado-a, entre otros.
- **Funciones:** son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología. Por ejemplo, pueden parecer funciones como categorizar_clase, asignar_fecha, entre otros.
- **Instancias:** se utilizan para representar objetos determinados de un concepto.

- Axiomas: son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Por ejemplo: “Si A y B son de la clase C, entonces A no es subclase de B”, “Para todo A que cumpla la condición C1, A es B”, entre otros.

Tecnologías de la Web semántica. XML (eXtensible Markup Language)

Es un lenguaje basado en etiquetas. Representa modelos de datos semi-estructurados.

Características:

- XML es flexible y es un metalenguaje.
- Las etiquetas pueden definirse.
- Adaptable a dominios/aplicaciones.
- Mejora a HTML en lenguaje dependiente del dominio.
- Correspondencias entre XML y HTML (XSL).
- Separa la estructura de los datos de la forma en que serán presentados.
- Permite auto-describir los datos.
- Estandariza el intercambio de datos y conocimiento.
- Permite la “integración” de datos con diferentes esquemas y propiedades.
- XML DTD’s o XMLSchema para definir la estructura del documento.
- Permite una validación mínima.

Limitaciones de XML:

- Muchas posibilidades diferentes de codificar un universo de discurso.
- No se dice nada sobre el significado de las estructuras.

- Difícil de entender otros documentos.
- Se debe entender el significado de cada vocabulario.
- No se logró una estandarización.

RDF (Resource Description Framework). Una opción para el manejo de grandes volúmenes de información puede ser el uso de metadatos para describir los datos contenidos en la Web. Los metadatos (por ejemplo, un catálogo de biblioteca es un registro de metadatos, en el sentido que describe publicaciones) son concretamente datos que describen recursos Web.

RDF (Infraestructura para la Descripción de Recursos) es una base para procesar metadatos; proporciona interoperabilidad entre aplicaciones que intercambian información legible por máquina en la Web. RDF destaca por la facilidad para habilitar el procesamiento automatizado de los recursos Web. Es un modelo de datos donde el Recurso es el ente del cual se habla, y la Propiedad es la que define relaciones del ente con otros entes o valores.

Recurso:

- Tiene Propiedad con Valor(Objeto).
- Recursos, Propiedades y Objetos están identificados con un único Universal Resource Identifier, URI (Berners-Lee, 2018).
- Las palabras en los documentos están atadas a una única definición que puede ser encontrada en la Web.

Propiedades de RDF:

- Existen clases primitivas
- Relaciones

- SubClassOf
- SubPropertyOf
- Instancias de Clases
- Existe una definición formal de la semántica de RDF
- Teoría de Modelos: Conjunto de Reglas de inferencia que permiten inferir nuevas tripletas

Desventajas:

- Dos documentos pueden usar definiciones diferentes del mismo concepto y no se puede decidir que son equivalentes.
- Tiene problemas de ambigüedad en la definición de sus elementos.
- No se permite el manejo/ definición de restricciones de integridad.

RDF Schema. El modelo y la sintaxis en RDF no facilitan los mecanismos para definir algunas propiedades ni las relaciones entre predicados y otros recursos o sujetos; por ello se precisó también una especificación para definir los esquemas. Así que en RDF schema (que es una extensión semántica de RDF), se hace uso de las clases; una clase es cualquier recurso que tenga un tipo (rdf:type) como propiedad cuyo valor sea un recurso rdfs:Class (Brickley y Guha 2014).

Entenderemos por la etiqueta "rdf:type" que todo recurso que la ocupe deberá tener las características que distinguen al recurso que se encuentre del lado derecho de la etiqueta.

Los elementos que pertenecen a las clases son conocidos como *instancias*. El sistema de propiedades y clases de RDF Schema (RDF(s)) es similar a los sistemas de tipos en lenguajes orientados a objetos como Java.

OWL. *OWL 2 Web Ontology Language* es un lenguaje basado en lógica, para la descripción de ontologías en la web semántica, creado como un estándar W3C, diseñado para representar conocimiento complejo acerca del mundo real: cosas, grupos de cosas y relaciones entre las cosas (Hitzler, Krötzsch, y Parsia, 2012). El conocimiento expresado en OWL puede ser razonado con programas de computador, sea para verificar la consistencia o para hacer explícito el conocimiento implícito.

Dentro de la sintaxis de OWL 2, los nombres son IRIs (*International Resource Identifiers*), identificadores internacionales de recursos. Como los IRIs son largos, a menudo se utilizan mecanismos de abreviación para escribirlos. La manera como se trabaja con tales abreviaciones es específica de cada formato que se use para codificar las ontologías en OWL (Hitzler, *et al.*, 2012).

En OWL el mismo axioma puede ser escrito en varias formas:

- Formato XML/RDF
- Formato de sintaxis funcional (OWL 2)
- Formato Manchester (Utilizado por los autores de Protègè¹, es más corto)
- Turtle (formato más conciso)

Las ventajas de OWL radican en la capacidad de las ontologías para ser escalables, compatibles con diferentes sistemas de información y extensibles.

Aprendizaje Ontológico para la Web Semántica. El aprendizaje ontológico facilita enormemente la construcción de ontologías por la ingeniería ontológica. Existe un número de disciplinas complementarias que alimentan diferentes tipos de datos sin estructuras,

¹ Protègè es una de las herramientas de edición de ontologías más utilizadas. <http://protege.stanford.edu/>

semiestructurados, y bien estructurados, para soportar un proceso de ingeniería ontológica semiautomático y cooperativo.

Los pasos para el aprendizaje de ontologías son los siguientes:

1. Las ontologías existentes son importadas y rehusadas a través de la migración de estructuras existentes o definiendo reglas de mapeo entre estas estructuras y la ontología que se establecerá.
2. En la fase de extracción ontológica se modela la mayor parte de la ontología objetivo, a partir del aprendizaje soportado en la alimentación desde documentos Web.
3. El bosquejo de la ontología objetivo necesita ser depurado para ajustar mejor la ontología a su propósito principal.
4. El refinamiento de la ontología aprovecha el dominio ontológico dado, pero completa la ontología en una granularidad fina (también en contraste a la extracción).
5. La aplicación objetivo principal sirve como una medida para validar la ontología resultante. Finalmente uno puede reiniciar el ciclo, por ejemplo, incluyendo nuevos dominios a la ontología construida o haciendo mantenimiento y actualizando su alcance.

Framework para el aprendizaje de ontologías. Para el caso de Maedche (2002) las ontologías primitivas comprenden:

- Un conjunto de cadenas que describe entradas léxicas \mathcal{L} para conceptos y relaciones.
- Un conjunto de conceptos \mathcal{C} .
- Una taxonomía de conceptos con múltiples herencias (heterarquía) \mathcal{H}_c .

- Un conjunto de relaciones no taxonómicas \mathcal{R} descrito por su dominio y rango de restricciones.
- Una heterarquía de relaciones, por ejemplo, un conjunto de relaciones taxonómicas $\mathcal{H}_{\mathcal{R}}$.
- Relaciones \mathcal{F} y \mathcal{G} que relacionan conceptos y relaciones con sus entradas léxicas, respectivamente.
- Un conjunto de axiomas \mathcal{A} que describen restricciones adicionales en la ontología y permite hacer hechos implícitos explícitos.

Esta estructura se acerca a los RDFs, con la diferencia que está explícita la consideración de entradas léxicas. La separación de referencia de concepto y denotación de concepto, la cual puede ser fácilmente expresada en RDF, permite proveer ontologías de dominio muy específico sin incurrir en un conflicto instantáneo cuando se mezclan ontologías (una solicitud estándar en la Web semántica).

El aprendizaje de ontologías recae en las estructuras ontológicas dadas y en los datos de entrada, para proponer nuevo conocimiento acerca de conceptos de interés razonable, relaciones, entradas léxicas, o acerca de enlaces entre esas entidades (proponiendo la adición, eliminación o fusión de algunos de ellos). Los resultados del proceso de aprendizaje ontológico se presentan al ingeniero ontológico en los gráficos del conjunto de resultados. El ingeniero puede entonces navegar por los resultados y decidir seguir, eliminar o modificar las propuestas en concordancia con el objetivo de su tarea.

Componentes para el aprendizaje de ontologías. Integrando las consideraciones anteriores en una arquitectura coherente para la extracción y mantenimiento de ontologías desde datos de la Web, se identifican varios componentes núcleo:

1. Un componente genérico de administración desarrollado con delegación de tareas y constituyendo la columna vertebral de la infraestructura.
2. Un componente de procesamiento de recursos, que trabaja con los datos de entrada de la Web, incluyendo, en particular, un sistema de procesamiento de lenguaje natural.
3. Una librería de algoritmos trabajando en la salida del componente anterior y en las estructuras ontológicas, que retorna conjuntos de resultados.
4. La interfaz gráfica de usuario para la ingeniería de ontologías, OntoEdit (Maedche, 2002).

RIF (Rule Interchange Format). Se presentan a continuación los dialectos de RIF:

RIF-BLD (RIF Basic Logic Dialect). Desde una perspectiva teórica, RIF-BLD corresponde al lenguaje de reglas Horn definidas con igualdad, y a una semántica estándar de primer orden. Sintácticamente, tiene un número de extensiones para soportar características como objetos y frames, como en F-logic, IRIs como identificadores para conceptos, y tipos de datos de XML esquema. Estas características hacen a RIF-BLD un lenguaje web-consciente.

RIF-PRD (RIF Production Rule Dialect). Es el dialecto que permite definir reglas de producción, las cuales pueden ser vistas como condición-acción, y son particularmente útiles para especificar comportamientos y soportar la separación entre lógica de negocios y objetos de negocios.

En la figura 3 se muestra el modelo de capas de la arquitectura de la web semántica con la posición de RIF y las aplicaciones ontológicas.

Grafárbol ontológico. Una ontología, con sus operaciones y restricciones se puede representar mediante una estructura de grafárbol ontológico. Para lo cual se utiliza la cualidad del Grafárbol, que le permite diferenciar tipos de nodos y tipos de relación (Chavarro, 2010).

Las relaciones utilizadas en una ontología se pueden describir por medio de una jerarquía de relaciones como la representada en la figura 4. En el Grafárbol ontológico, sólo son representadas relaciones dirigidas, las cuales pueden ser jerárquicas (EsUn) o Funcionales. En el conjunto de las relaciones funcionales se destaca un subconjunto: Las relaciones de partes (ParteDe).

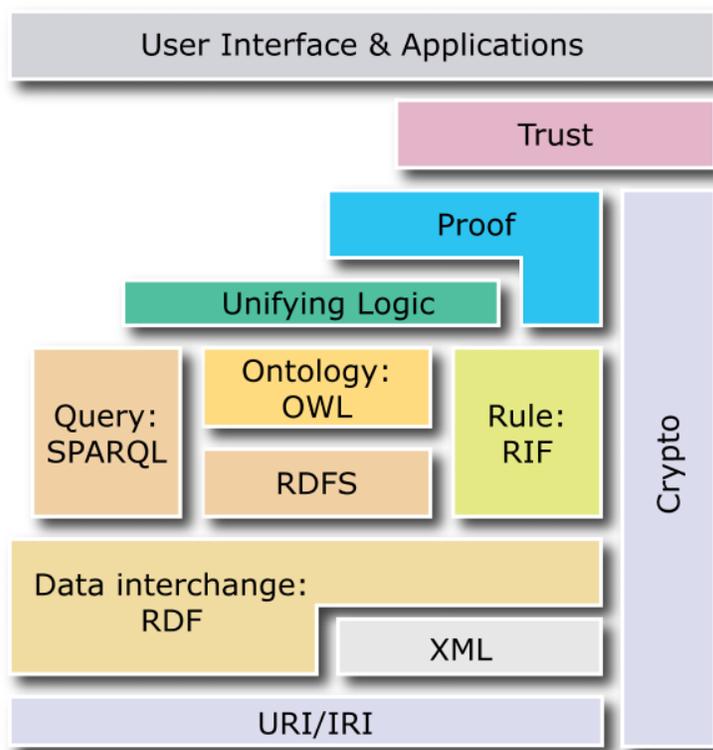


Figura 3. Arquitectura de la web semántica. Basado en Herman (2007)

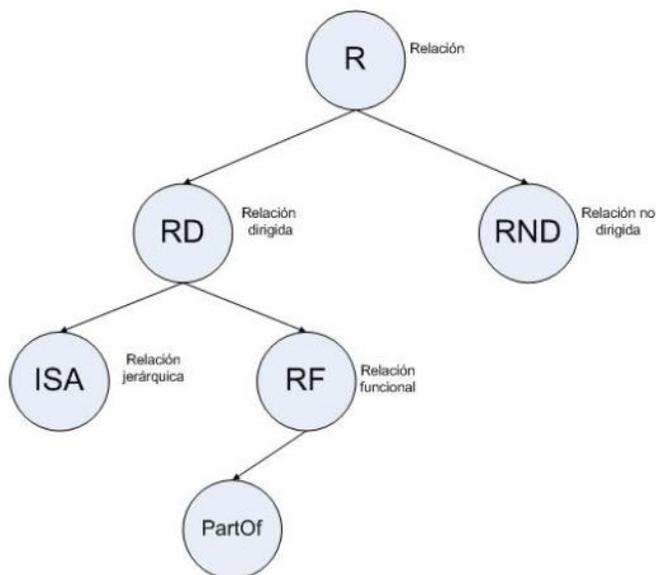


Figura 4: Jerarquía de las Relaciones. Basado en Chavarro (2010)

Un grafárbol ontológico está relacionado con la definición de ontología (dada en Chavarro, 2010) de la siguiente forma:

Los conceptos y los tipos de datos, se representan usando NodosConcepto C. Los Individuos y los literales se representan mediante NodosIndividuo I, Las relaciones entre conceptos, entre conceptos e individuos, entre individuos, entre conceptos y tipos de datos, o entre individuos y tipos de datos o valores, se representan usando Arcos A.

Cuando se diferencian los tipos de nodos, se puede conformar dos niveles en el Grafárbol: el nivel de términos, o nivel terminológico, y el nivel de instancias o individuos. El nivel de instancias, conforma un grafo a partir de las relaciones que se pueden describir entre los individuos o entre los individuos y los valores. Este nivel está interrelacionado con el nivel de términos por relaciones jerárquicas. El nivel terminológico proporciona una estructura de árbol conformado por la relación jerárquica.

Ontología de cambio genérico. De acuerdo con Palma (2009), la representación del cambio en una ontología es una de las tareas identificadas por la mayoría de los enfoques de evolución ontológica y se refiere a la representación de una solicitud de cambio formalmente y explícitamente como uno o más cambios ontológicos (Stojanovic, 2004). Los enfoques para la representación de los cambios son dependientes del modelo ontológico subyacente y consideran los cambios a nivel de entidad (conceptos, propiedades, individuos).

En Palma (2009) se propone un modelo de capas para la representación de los cambios ontológicos que integra muchas de las características de las ontologías de cambio existentes. Este modelo consiste en una *Ontología De Cambio Genérico*, independiente del modelo ontológico subyacente, que modela operaciones genéricas en una taxonomía de cambios que se espera sean compatibles con cualquier lenguaje de ontologías.

La ontología de cambio genérico define algunas de sus propiedades con un rango sin restricción, para evitar dependencias en un lenguaje ontológico específico. Así, en el enfoque en capas la ontología de cambio genérico puede ser especializada para lenguajes de ontologías específicos, al tiempo que proporciona un modelo común e independiente para la representación de los cambios en la ontología.

Las clases principales y propiedades de la Ontología de cambio genérico se ilustran en la figura 5.

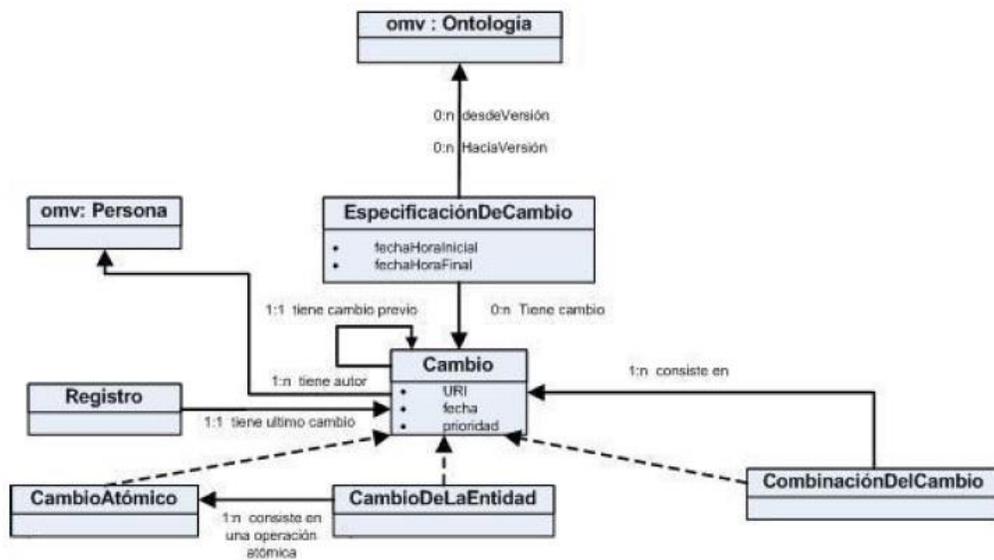


Figura 5. Clases principales de la Ontología de Cambio Genérico. Basado en Chavarro (2010).

Capítulo 5

Desarrollo de la propuesta

5.1. Análisis de la capacidad expresiva de RIF

En este apartado se abordarán los conceptos generales sobre RIF, y específicamente, del dialecto lógico básico BLD (Basic Logic Dialect) de RIF, que en adelante llamaremos RIF-BLD. Se realiza un análisis de la expresividad del lenguaje, teniendo en cuenta las características factibles de ser utilizadas para los propósitos del presente proyecto.

Generalidades

Como se explicó anteriormente, RIF es una familia de estándares que especifican varios dialectos, de acuerdo con el diagrama de la figura 6.

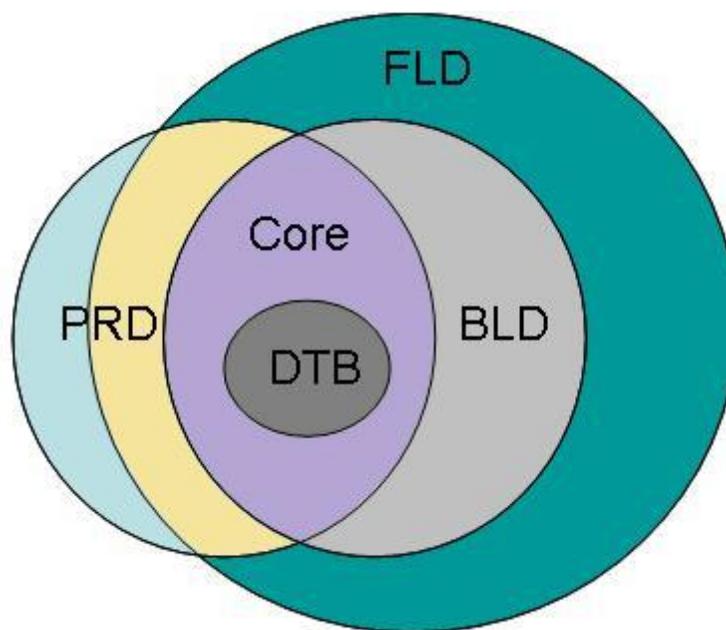


Figura 6. Familia de dialectos RIF. Basado en Paschke, Morgenstern, Hirtle, Ginsberg, Patranjan, y McCabe (2013)

Cada dialecto es una colección de componentes que trabajan juntos, formando un *interlingua*. Se necesitan nuevos dialectos para aquellos casos en los que ningún dialecto existente proporciona las características de lenguaje de reglas requeridas para el intercambio (Paschke, *et al.*, 2013).

El objetivo de creación de RIF fue el determinar una infraestructura de lenguaje estándar que soporte la interoperabilidad entre sistemas de reglas en la web, además de promover coherencia entre diferentes aplicaciones o capas, mientras se preservan sus requisitos específicos.

Como especifica la W3C, RIF pretende describir relaciones en formato de reglas, pues hay casos en que la lógica descriptiva (Subyacente a OWL, entre otros lenguajes de descripción de ontologías) no llega a cierta capacidad de razonamiento. Pero no es una tarea fácil, pues sigue un paradigma más estricto a la hora de representar el conocimiento (denominado “Close World Assumption” o “Presunción de Mundo Cerrado”) que el otro paradigma (denominado “Open World Assumption” o “Presunción de Mundo Abierto”) (Bertino, 2003), que es más adecuado para la información dispersa en la web. Actualmente, OWL y RIF se desarrollan en paralelo.

En (Higuera, 2010) se da un ejemplo muy sencillo de la esencia de RIF:

“Si dos clínicas médicas deciden publicar en Internet la lista de sus trabajadores etiquetándolos por sus apellidos, puede suceder que usen dos palabras diferentes para categorizarlos: “surname” y “last name”. Si un usuario busca en Internet los médicos de su ciudad y escribe uno de esos dos sustantivos, es muy probable que no encuentre a los profesionales etiquetados con la otra palabra utilizada para referirse al apellido.

Para resolver este problema ya existe OWL, que en el anterior ejemplo proporcionaría a los

desarrolladores una manera de identificar que “surname”, “last name”, y quizá, “family name” hacen referencia al mismo tipo de datos. Pero, ¿qué ocurriría si una tercera clínica decidiera unir el nombre, el apellido y la inicial del segundo apellido en una sola categoría denominada *nombre*? De nuevo, esta regla no funcionaría. Sería necesaria una norma que unificase los datos.

Ahora, los investigadores del MIT, que trabajan para el W3C, creen haberla encontrado. Se llama RIF, y se trata de un estándar que ofrece a los desarrolladores web una nueva forma de escribir normas para intercambiar datos de diferentes sitios de Internet de manera exitosa”.

De esta manera, se visualiza RIF como la solución a las limitaciones de OWL, sobre todo en lo que tiene que ver con la representación del conocimiento en formato de reglas declarativas y de producción.

Dialectos RIF. ¿Cómo puede un motor de reglas trabajar con conjuntos de reglas de diferentes lenguajes? con esta pregunta nos acercamos al diseño de RIF, que refleja la lógica de identificar tipos específicos de reglas dentro de sistemas de reglas existentes, llamados Dialectos RIF, que pueden ser traducidos a otros sistemas de reglas sin cambiar su significado:

- RIF-core: el lenguaje RIF básico. Todos los dialectos de RIF son una extensión de RIF-Core + DTB (Morgenstern y Welty, 2013).
- DTB: un conjunto de tipos de datos y funciones incorporadas y predicados que pueden ser usados cuando escribimos reglas.

- RIF-BLD: RIF Basic Logic Dialect. RIF-BLD tiene el RIF-Core dialect como un subconjunto.
- RIF-PRD (RIF Production Rule Dialect): Es el dialecto que permite definir reglas de producción, las cuales pueden ser vistas como condición-acción.

RIF-BLD. Es el dialecto RIF que se toma como base y delimitación para la realización de la extensión del grafárbol, objeto de la presente propuesta.

Es el dialecto de RIF utilizado para escribir reglas lógicas. Desde una perspectiva teórica, RIF-BLD corresponde al lenguaje de reglas de Horn² definidas con igualdad, y a una lógica estándar de primer orden (Boley y Kifer, 2013).

Sintácticamente, tiene un número de extensiones para soportar características como objetos y frames, como en F-logic, IRIs como identificadores para conceptos, y tipos de datos de XML esquema. Estas características hacen a RIF-BLD un lenguaje web-consciente (web-aware). Se debe recordar que RIF está diseñado para permitir la interoperabilidad entre lenguajes de reglas en general, y sus usos no están limitados a la web.

No se espera que sea posible traducir a RIF-BLD todas las reglas de la mayoría de sistemas de reglas, al contrario, se espera que solo cierta clase de reglas se puedan traducir. RIF comparte ciertas características con ISO Common Logic (CL), en sí mismo una evolución de KIF y grafos conceptuales.

² Cláusula de Horn: es una cláusula con a lo sumo un literal positivo, es decir, innegado.

Contrario a CL, RIF-BLD fue diseñado como un dialecto simple con expresividad limitada que recae dentro de la intersección entre sistemas de programación lógica y de primer orden. Por eso BLD no soporta la negación. Common Logic es estrictamente de primer orden. Para intercambio de reglas entre CL y dialectos RIF, se espera que se definan mapeos parciales RIF-CL (Boley y Kifer, 2013).

Los dialectos RIF basados en lógica, que especializan o extienden a BLD de acuerdo con el FLD (RIF Framework for Logic Dialects RIF-FLD), incluyen el RIF-core como una especialización de BLD. Se espera que otras especificaciones desarrollen más dialectos lógicos basados en BLD, por ejemplo, una extensión de BLD que capture incertidumbre.

RIF-BLD también tiene algo de similaridad con SPARQL, en particular con respecto a la compatibilidad RDF. Al igual que con la correspondencia entre un fragmento de SQL y Datalog, SPARQL puede ser mapeado parcialmente a Datalog (y así al subconjunto RIF-Core de RIF-BLD). Un mapeo completo de SPARQL necesitaría constructos más allá de RIF-BLD, tal como la negación no-monótona.

RIF-BLD se define en dos diferentes formas, ambas normativas (Boley y Kifer, 2013):

- Como una especificación directa, independientemente del framework RIF-FLD, para beneficio de los que desean un camino directo a RIF-BLD, por ejemplo, implementadores potenciales, y que no están interesados en temas de extensibilidad.
- Como especialización de RIF-FLD: Es útil para los que ya están familiarizados con el framework y no necesitan la especificación directa de RIF-BLD; también es útil para diseñadores de dialectos.

Sintaxis de RIF-BLD

A continuación se describen ejemplos de código en RIF-BLD, para examinar los elementos característicos y la semántica implícita.

RIF-BLD usa una sintaxis compacta llamada sintaxis de presentación. En el siguiente ejemplo se observa dicha sintaxis, a la vez que se hace una introducción a la definición de una regla:

- **A buyer buys an item from a seller if the seller sells the item to the buyer** (*un comprador compra un ítem de un vendedor si el vendedor vende el ítem al comprador*).
- **John sells LeRif to Mary** (*John vende LeRif a Mary*).

```
Document (
  Base(<http://example.com/people#>)
  Prefix(cpt <http://example.com/concepts#>)
  Prefix(bks <http://example.com/books#>)
  Group
    (
      Forall ?Buyer ?Item ?Seller (
        cpt:buy(?Buyer ?Item ?Seller) :- cpt:sell(?Seller
          ?Item ?Buyer)
      )
      cpt:sell(<John> bks:LeRif "Mary"^^rif:iri)
    )
)
```

Hasta aquí, y haciendo uso de un razonador apropiado, se debe inferir que “Mary compra LeRif a John”. Obsérvese la notación infija en la implicación (Rule implication) ‘:-’

Siempre que un documento RIF-BLD caiga en el subconjunto Core o pueda ser traducido a éste, el documento debe ser producido en RIF-core para permitir su intercambio con un máximo número de consumidores RIF. Por ejemplo, el documento del ejemplo anterior, que se parece a

Datalog, también es un ejemplo RIF-Core. Puesto que RIF es un formato de intercambio, usa XML como la única sintaxis concreta.

A continuación otro ejemplo:

```
Document (
  Prefix(rdfs <http://www.w3.org/2000/01/rdf-schema#>)
  Prefix(imdbrel <http://example.com/imdbrelations#>)
  Prefix(dbpedia <http://dbpedia.org/ontology/>)

  Group (
    Forall ?Actor ?Film ?Role (
      If And(imdbrel:playsRole(?Actor ?Role)
imdbrel:roleInFilm(?Role ?Film))
      Then dbpedia:starring(?Film ?Actor)
    )
  )
)
```

Nótese la sentencia Group (usada como conjunción, es decir que funciona agregando fórmulas RIF-BLD en colecciones), y la sentencia “And” dentro de la formulación de la implicación (aquí la sintaxis es de RIF-core, aunque el significado es igual que en ‘:-’).

Ejemplo 3.

```
ex:john ex:brotherOf ex:jack .
ex:jack ex:parentOf ex:mary .
```

UncleOf

```
Forall ?x ?y ?z (?x[ex:uncleOf -> ?z] :-
  And(?x[ex:brotherOf -> ?y] ?y[ex:parentOf -> ?z]))
```

Figura 7. Ejemplo de sintaxis RIF-BLD. Basado en Lera (2014).

En este ejemplo, la regla *TíoDe* se expresa como implicación de la conjunción de *SerHermano* y *SerPadre*, para todas las variables que cumplan con la condición.

Capacidad expresiva de RIF-BLD

Después de observar los ejemplos, haciendo una recopilación de toda la sintaxis de RIF-BLD, y teniendo en cuenta que su especialidad es la expresividad de reglas en un formato estándar, se desarrolla un análisis de las características básicas del dialecto, el cual se resume en la tabla 1.

Otras observaciones.

- El cuantificador universal aplica en dos formas distintas, para reglas universales y para hechos universales. Esta segunda forma se refiere a la aplicación de fórmulas atómicas (que no son implicaciones).
- La membresía # también es expresada de la forma
`rdf:type (a B)`
 Donde se especifica que a es un individuo de la clase B (teniendo en cuenta la sintaxis XML).
- Los argumentos nombrados o frames se aplican de manera similar a un arreglo, es decir, expresando los componentes como en un vector.

Expresión de presentación RIF-BLD	Observación
Const	Declaración de constantes
Var	Declaración de variables
ArgNames	Arreglo de nombres
And	Conjunción
Or	Disyunción
:-	Implicación (<i>Rule implication</i>)
Exists	Cuantificador existencial
Forall	Cuantificador universal
=	Igualdad
#	Membresía (Membership)
##	Subclase
->	Argumentos nombrados o frames
External	Un argumento definido en otro documento
Import	
Prefix	
Base	Prefix y Base habilitan representaciones compactas de IRIs
Group	Maneja fórmulas RIF-BLD como colecciones. Se trata como una conjunción (And).
Document	
List	Listas
OpenList	Lista abierta

Tabla 1. Expresiones en RIF-BLD en su sintaxis de presentación

- Las listas (*List* y *OpenList*) se desarrollan de manera similar al manejo que se les da en Common LISP (Steele, 1990).

Después de revisar las características de expresividad de RIF-BLD, se identifican algunos elementos base que se pueden tener en cuenta para utilizarlos en la implementación del grafárbol:

- La característica principal y determinante de RIF (y en nuestro caso de RIF-BLD) es la expresión de reglas declarativas. RIF-BLD usa la sintaxis de *Rule Implication* (‘:-’ o sentencias if-then) para denotar una regla (un hecho sobre algo del mundo). Esto significa que muchas sentencias de RIF-BLD (predominantemente reglas) no pueden ser expresadas como parte de un documento OWL.
- Los cuantificadores existencial y universal tienen un uso ligeramente diferente: se utilizan dentro de la declaración de una regla, lo que en el presente objeto de estudio no correspondería a la sintaxis de OWL y por tanto del lenguaje declarativo (el conjunto de operadores de cambio que implementan tecnológicamente al grafárbol).
- RIF-BLD no permite la negación (*not*). Para expresar negación se utiliza el dialecto PRD (Production Rule Dialect) de RIF, el cual no es objeto de la presente propuesta.
- El concepto *List* para denotar objetos agrupados en listas, tanto abiertas como cerradas, es similar al concepto en Common LISP.

En el siguiente apartado se revisará la semántica de las expresiones en RIF-BLD, con base en los elementos analizados previamente, para precisar cuáles de estas características son útiles en la extensión del grafárbol, o dicho de otra manera, cuáles elementos son nuevos a OWL.

5.2. Semántica de RIF-BLD

Después de revisar toda la sintaxis de RIF-BLD, es necesario elaborar una comparativa con los lenguajes de representación de ontologías, lo que permitirá establecer unas sentencias base de RIF que puedan aportar a la extensión del grafárbol ontológico, objeto de estudio del presente proyecto. En este apartado se presenta la semántica del dialecto y el resumen de la correspondencia con OWL, a partir del cual se obtienen las sentencias “nuevas” para ser tratadas.

Semántica del dialecto

A continuación se explica la semántica directa del RIF-Basic logic Dialect (Boley y Kifer, 2013):

Hay que tener en cuenta que la sintaxis de RIF-BLD que se ha usado hasta el momento es la de presentación, es decir que permite la notación abreviada, que se especifica a través de las directivas Prefijo y Base, y varios atajos para enteros, cadenas y *rif*: símbolos locales. La semántica se describe utilizando la sintaxis completa, es decir, se supone que todos los accesos directos ya se han expandido como se define en (Polleres, Boley y Kifer, 2013). Esta es una semántica directa, lo que significa que se especifica sin depender del framework FLD (Boley y Kifer, 2013b).

1. Valores de verdad

El conjunto TV de valores de verdad en RIF-BLD consta de dos valores, t y f .

2. Estructuras semánticas

El concepto clave en una semántica de teoría de modelos para un lenguaje lógico es la noción de una *estructura semántica* (Enderton, 2001). La definición es un poco más general que lo estrictamente necesario para RIF-BLD. Esto sienta las bases para las extensiones de RIF-BLD y hace que la conexión con la semántica del framework FLD sea más obvia.

Definición (estructura semántica). Una *estructura semántica*, I , es una tupla de la forma

$$\langle TV, DTS, D, D_{ind}, D_{func}, I_C, I_V, I_F, I_{NF}, I_{list}, I_{tail}, I_{frame}, I_{sub}, I_{isa}, I_{=}, I_{external}, I_{truth} \rangle$$

Donde D es un conjunto no vacío de elementos llamado el *dominio* de I , y D_{ind} , D_{func} son subconjuntos no vacíos de D . D_{ind} se usa para interpretar los elementos de $Const$ que ocurren como individuos y D_{func} se usa para interpretar los elementos de $Const$ que ocurren en el contexto de los símbolos funcionales.

$Const$ denota el conjunto de todos los símbolos constantes y Var el conjunto de todos los símbolos variables. DTS denota un conjunto de identificadores para tipos de datos.

Los otros componentes de I son mapeos totales definidos de la siguiente manera:

1. I_C asigna $Const$ a D .

Este mapeo interpreta símbolos constantes. Además:

- Si una constante, $c \in Const$, es un *individuo*, entonces se requiere que

$$I_C(c) \in D_{ind}$$

- Si $c \in \text{Const}$, es un *símbolo de función* (posicional o con argumentos nombrados), entonces se requiere que $I_C(c) \in D_{func}$.

2. I_V mapea Var a D_{ind} .

Este mapeo interpreta símbolos variable.

3. I_F asigna D a las funciones totales $D^*_{ind} \rightarrow D$

(Aquí D^*_{ind} es un conjunto de todas las secuencias finitas sobre el dominio D_{ind}).

Este mapeo interpreta términos posicionales. Además:

- Si $d \in D_{func}$ entonces $I_F(d)$ debe ser una función $D^*_{ind} \rightarrow D_{ind}$.
- Esto significa que cuando se aplica un símbolo de función a argumentos que son objetos individuales, el resultado también es un objeto individual.

4. I_{NF} mapea D al conjunto de funciones totales de la forma

$$\text{SetOfFiniteSets}(\text{ArgNames} \times D_{ind}) \rightarrow D$$

Este mapeo interpreta símbolos de funciones con argumentos nombrados. Además:

- Si $d \in D_{func}$, entonces $I_{NF}(d)$ debe ser una función $\text{SetOfFiniteSets}(\text{ArgNames} \times D_{ind}) \rightarrow D_{ind}$.
- Esto es análogo a la interpretación de términos posicionales con dos diferencias:
 - ✓ Cada par $\langle s, v \rangle \in \text{ArgNames} \times D_{ind}$ representa un par argumento/valor en lugar de solo un valor en el caso de un término posicional.
 - ✓ Los argumentos de un término con argumentos nombrados constituyen un conjunto finito de pares argumento/valor en lugar de una secuencia ordenada finita de simples elementos. Entonces, el orden de los argumentos no importa.

5. I_{list} e I_{tail} se usan para interpretar listas. Son mapeos de la forma:

- $I_{list}: D_{ind}^* \rightarrow D_{ind}$
- $I_{tail}: D_{ind}^+ \times D_{ind} \rightarrow D_{ind}$

Además, estas asignaciones deben cumplir las siguientes condiciones:

- La función I_{list} es inyectiva (uno a uno).
- El conjunto $I_{list}(D_{ind}^*)$, denominado en adelante D_{list} , está separado de los espacios de valores de todos los tipos de datos en DTS .
- $I_{tail}(a_1, \dots, a_k, I_{list}(a_{k+1}, \dots, a_{k+m})) = I_{list}(a_1, \dots, a_k, a_{k+1}, \dots, a_{k+m})$.

Nótese que la última condición anterior restringe I_{tail} solo cuando su último argumento está en D_{list} . Si el último argumento de I_{tail} no está en D_{list} , entonces la lista es una abierta general y no hay restricciones en el valor de I_{tail} , excepto que debe estar en D_{ind} .

6. Mapas $I_{frame} D_{ind}$ a funciones totales de la forma:

$$\text{SetOfFiniteBags}(D_{ind} \times D_{ind}) \rightarrow D.$$

Este mapeo interpreta términos *frame*. Un argumento, $d \in D_{ind}$, a I_{frame} representa un objeto y la bolsa finita $\{ \langle a_1, v_1 \rangle, \dots, \langle a_k, v_k \rangle \}$ representa una bolsa de pares atributo-valor para d . Ejemplo:

Los paquetes (conjuntos múltiples) se usan aquí porque el orden de los pares atributo / valor en un frame es inmaterial y los pares pueden repetirse. Tales repeticiones surgen naturalmente cuando las variables son instanciadas con constantes. Por ejemplo,

$\circ[?A \rightarrow ?B \ ?C \rightarrow ?D]$ se convierte en $\circ[a \rightarrow b \ a \rightarrow b]$ si las variables $?A$ y $?C$ se instancian con el símbolo a mientras $?B$ y $?D$ son instanciadas con b . (También veremos que $\circ[a \rightarrow b \ a \rightarrow b]$ es equivalente a $\circ[a \rightarrow b]$).

7. I_{sub} le da significado a la relación de subclase. Es un mapeo de la forma

$$D_{ind} \times D_{ind} \rightarrow D.$$

I_{sub} se restringirá más adelante en el numeral Interpretación de fórmulas, para garantizar que el operador $\#\#$ sea transitivo, es decir, que $c1\#\#c2$ y $c2\#\#c3$ implican $c1\#\#c3$.

8. I_{isa} le da significado a la membresía de clase (*class membership*). Es un mapeo de la forma $D_{ind} \times D_{ind} \rightarrow D$.

I_{isa} se restringirá aún más en el numeral Interpretación de fórmulas para garantizar que las relaciones $\#$ y $\#\#$ tengan la propiedad habitual que todos los miembros de una subclase también sean miembros de la superclase, es decir, que $o\#c1$ y $c1\#\#scl$ impliquen $o\#scl$.

9. $I_{=}$ es un mapeo de la forma $D_{ind} \times D_{ind} \rightarrow D$.

Da sentido al operador de igualdad.

10. I_{truth} es un mapeo de la forma $D \rightarrow TV$.

Se usa para definir el valor de verdad para fórmulas.

11. $I_{external}$ es un mapeo del conjunto coherente de esquemas para funciones definidas externamente a funciones totales $D^* \rightarrow D$.

Para cada esquema externo $\sigma = (?X_1 \dots ?X_n; \tau)$ en el conjunto coherente de esquemas externos asociados con el lenguaje, $I_{external}(\sigma)$ es una función de la forma $D^n \rightarrow D$.

Para cada esquema externo, σ , asociado con el lenguaje, se asume que $I_{external}(\sigma)$ se especifica externamente en algún documento (de ahí el nombre de *esquema externo*). En particular, si σ es un esquema de un predicado o función incorporada de RIF, $I_{external}(\sigma)$ se especifica en (Polleres, Boley, y Kifer, 2013).

También se define el siguiente mapeo de términos a D , que denotamos usando el mismo símbolo I como el usado para estructuras semánticas. Esta sobrecarga es conveniente y no crea ambigüedad.

- $I(k) = I_C(k)$, si k es un símbolo en Const
- $I(?V) = I_V(?V)$, si $?V$ es una variable en Var
- $I(f(t_1 \dots t_n)) = I_F(I(f))(I(t_1), \dots, I(t_n))$
- $I(f(s_1 \rightarrow v_1 \dots s_n \rightarrow v_n)) = I_{NF}(I(f))(\{ \langle s_1, I(v_1) \rangle, \dots, \langle s_n, I(v_n) \rangle \})$

Aquí usamos $\{ \dots \}$ para denotar un conjunto de pares argumento / valor.

- Para los términos *list*, la asignación se define de la siguiente manera:

$$\checkmark I(List()) = I_{list}(\langle \rangle).$$

Aquí $\langle \rangle$ denota una lista vacía de elementos de D_{ind} . (Nótese que el dominio de I_{list} es

D_{ind}^* , por lo que D_{ind}^0 es una lista vacía de elementos de D_{ind}).

$$\checkmark I(List(t_1 \dots t_n)) = I_{list}(I(t_1), \dots, I(t_n)), \text{ si } n > 0.$$

$$\checkmark I(List(t_1 \dots t_n | t)) = I_{tail}(I(t_1), \dots, I(t_n), I(t)), \text{ si } n > 0.$$

- $I(o[a_1 \rightarrow v_1 \dots a_k \rightarrow v_k]) = I_{frame}(I(o))(\{ \langle I(a_1), I(v_1) \rangle, \dots, \langle I(a_k), I(v_k) \rangle \})$

Aquí $\{...\}$ denota un grupo de pares atributo / valor. Los elementos duplicados en un grupo de este tipo no afectan el valor de verdad de una fórmula frame. Por lo tanto, por ejemplo, $\circ[a \rightarrow b \ a \rightarrow b]$ y $\circ[a \rightarrow b]$ siempre tienen el mismo valor de verdad.

- $I(c1 \# \# c2) = I_{sub}(I(c1), I(c2))$
- $I(o \# c) = I_{isa}(I(o), I(c))$
- $I(x = y) = I_{=} (I(x), I(y))$
- $I(\text{External}(\tau)) = I_{external}(\sigma)(I(s_1), \dots, I(s_n))$, si τ es una instanciación del esquema externo $\sigma = (?X_1 \dots ?X_n; \tau)$ por sustitución $?X_1/s_1 \dots ?X_n/s_n$.

Nótese que, por definición, $\text{External}(\tau)$ está bien formado solo si τ es una instanciación de un esquema externo. Además, mediante la definición de conjuntos coherentes de esquemas externos, τ puede ser una instancia de como máximo uno de dichos esquemas, por lo que $I(\text{External}(\tau))$ está bien definido.

Hay que tener en cuenta que las definiciones de I_{NF} e $I(x = y)$ implican que los términos con argumentos nombrados que difieren sólo en el orden de sus argumentos, son mapeados por I al mismo elemento en el dominio. Esto implica que las igualdades como

$$t(a \rightarrow 1 \ b \rightarrow 2 \ c \rightarrow 3) = t(c \rightarrow 3 \ a \rightarrow 1 \ b \rightarrow 2)$$

Son tautologías en RIF-BLD.

El efecto de los tipos de datos. El conjunto DTS debe incluir los tipos de datos descritos dentro del estándar. Los identificadores de tipo de datos en DTS imponen las siguientes restricciones:

Dado $dt \in DTS$, LS_{dt} denotará el espacio léxico de dt , VS_{dt} denota su espacio de valor, y $L_{dt}: LS_{dt} \rightarrow VS_{dt}$ el mapeo léxico a espacio de valor. Entonces se debe cumplir lo siguiente:

- $VS_{dt} \subseteq D_{ind}$; y
- Para cada constante " lit " ^{dt} tal que $lit \in LS_{dt}$, $I_C("lit" ^{$dt$}) = $L_{dt}(lit)$.$

Esto es, I_C debe mapear las constantes de un tipo de datos dt de acuerdo con L_{dt} .

RIF-BLD no impone restricciones en I_C para constantes en espacios de símbolos que no son tipos de datos incluidos en DTS .

3. Anotaciones de RIF-BLD en la Semántica

Las anotaciones en RIF-BLD se eliminan antes de que se apliquen los mapeos que constituyen las estructuras semánticas de RIF-BLD. De la misma manera, se eliminan antes de aplicar el valor de verdad, $TVal_I$, definido más adelante. Por lo tanto, los identificadores y los metadatos no tienen ningún efecto sobre la semántica formal.

Se observa que aunque los identificadores y los metadatos asociados con fórmulas RIF-BLD son ignorados por la semántica, se pueden extraer mediante herramientas XML. Los términos frame utilizados para representar metadatos en RIF-BLD se pueden alimentar a otras reglas RIF-BLD, lo que permite el razonamiento sobre los metadatos. Sin embargo, RIF-BLD no define ninguna semántica particular para metadatos (Boley y Kifer, 2013).

Aquí cabe mencionar que dentro del trabajo de (Chavarro, 2010) no se encuentran implementados los operadores de anotaciones, los cuales no han sido incorporados en el lenguaje declarativo. Esto significa que para extender el grafábol ontológico con sentencias propias de RIF, no tendremos en cuenta estos operadores.

4. Interpretación de fórmulas no documentales

Se define cómo una estructura semántica, I , determina el valor de verdad $TVal_I(\varphi)$ de una fórmula RIF-BLD, φ , donde φ es cualquier fórmula que no sea de documento. La valoración de verdad de fórmulas documentales se define en el siguiente punto.

Definimos un mapeo, $TVal_I$, del conjunto de todas las fórmulas no documentales a TV . Nótese que la definición implica que $TVal_I(\varphi)$ se define solo si el conjunto DTS de los tipos de datos de I incluye todos los tipos de datos mencionados en φ e I_{external} se define en todas las funciones y predicados definidos externamente en φ .

Definición (Valor de verdad). El *valor de verdad* para fórmulas bien formadas en RIF-BLD se determina usando la siguiente función, denominada $TVal_I$:

1. Fórmulas atómicas posicionales: $TVal_I(r(t_1 \dots t_n)) = I_{\text{truth}}(I(r(t_1 \dots t_n)))$

2. Fórmulas atómicas con argumentos nombrados:

$$TVal_I(p(s_1 \rightarrow v_1 \dots s_k \rightarrow v_k)) = I_{\text{truth}}(I(p(s_1 \rightarrow v_1 \dots s_k \rightarrow v_k))).$$

3. Igualdad: $TVal_I(x = y) = I_{\text{truth}}(I(x = y))$.

- Para garantizar que la igualdad tenga exactamente las propiedades esperadas, se requiere que:

$$\checkmark I_{\text{truth}}(I(x = y)) = t \text{ si } I(x) = I(y) \text{ e } I_{\text{truth}}(I(x = y)) = f \text{ de lo contrario.}$$

$$\checkmark \text{ Esto es equivalente a decir que } TVal_I(x = y) = t \text{ si y solo si } I(x) = I(y).$$

4. Subclase: $TVal_I(sc \#\# c1) = I_{\text{truth}}(I(sc \#\# c1))$.

Para garantizar que el operador $\#\#$ sea transitivo, se requiere lo siguiente:

- Para todos los términos bien formados $c1, c2, c3$: si

$$TVal_I(c1 \#\# c2) = TVal_I(c2 \#\# c3) = t \text{ entonces } TVal_I(c1 \#\# c3) = t.$$

5. Membresía: $TVal_I(o \# c_l) = I_{truth}(I(o \# c_l))$.

Para garantizar que todos los miembros de una subclase también sean miembros de la superclase, se requiere:

- Para todos los términos bien formados o , c_l , scl :

Si $TVal_I(o \# c_l) = TVal_I(c_l \# scl) = t$, entonces $TVal_I(o \# scl) = t$.

6. Frame: $TVal_I(o [a_1 \rightarrow v_1 \dots a_k \rightarrow v_k]) = I_{truth}(I(o [a_1 \rightarrow v_1 \dots a_k \rightarrow v_k]))$.

Como el conjunto de pares atributo/valor asociados con un objeto o representa la conjunción de aserciones representadas por estos pares, se requiere, si $k > 0$:

- $TVal_I(o [a_1 \rightarrow v_1 \dots a_k \rightarrow v_k]) = t$ si y solo si

$TVal_I(o [a_1 \rightarrow v_1]) = \dots = TVal_I(o [a_k \rightarrow v_k]) = t$.

7. Fórmula atómica definida externamente:

$TVal_I(External(\tau)) = I_{truth}(I_{external}(\sigma)(I(s_1), \dots, I(s_n)))$, si τ es una fórmula atómica que es una instanciación del esquema externo $\sigma = (?X_1 \dots ?X_n; \tau)$ por sustitución $?X_1/s_1 \dots ?X_n/s_n$.

Nótese que, por definición, $External(\tau)$ está bien formado solo si τ es una instancia de un esquema externo. Además, mediante la definición de conjuntos coherentes de esquemas externos, τ puede ser una instancia de como máximo uno de dichos esquemas, así que $I(External(\tau))$ está bien definido.

8. Conjunción: $TVal_I(And(c_1 \dots c_n)) = t$ si y solo si $TVal_I(c_1) = \dots = TVal_I(c_n) = t$.

De lo contrario, $TVal_I(And(c_1 \dots c_n)) = f$.

La conjunción vacía es tratada como una tautología, por lo que $TVal_I(And()) = t$.

9. Disyunción: $TVal_I(Or(c_1 \dots c_n)) = f$ si y solo si $TVal_I(c_1) = \dots = TVal_I(c_n) = f$.

De lo contrario, $TVal_I(Or(c_1 \dots c_n)) = t$.

La disyunción vacía se trata como una contradicción, por lo que $TVal_I(\text{OR}()) = f$.

10. Cuantificación:

- $TVal_I(\text{Exists } ?V_1 \dots ?V_n (\varphi)) = t$ si y solo si para algún I^* , descrito a continuación, $TVal_I^*(\varphi) = t$.
- $TVal_I(\text{Forall } ?V_1 \dots ?V_n (\varphi)) = t$ si y solo si para cada I^* , $TVal_I^*(\varphi) = t$.

Aquí I^* es una estructura semántica de la forma

$\langle TV, DTS, D, D_{ind}, D_{func}, IC, I^*_V, IF, INF, I_{list}, I_{tail}, I_{frame}, I_{sub}, I_{isa}, I_=, I_{external}, I_{truth} \rangle$

Que es exactamente como I , excepto que se usa el mapeo I^*_V en lugar de I_V . I^*_V se define para que coincida con I_V en todas las variables excepto, posiblemente, en $?V_1, \dots, ?V_n$.

11. Implicación (*rule implication*):

- $TVal_I(\text{conclusión} :- \text{condición}) = t$, si $TVal_I(\text{conclusión}) = t$ o $TVal_I(\text{condición}) = f$.
- $TVal_I(\text{conclusión} :- \text{condición}) = f$ de lo contrario.

12. Grupos de reglas:

Si Γ es una fórmula *Group* de la forma $\text{Group}(\varphi_1 \dots \varphi_n)$, entonces

- $TVal_I(\Gamma) = t$ si y solo si $TVal_I(\varphi_1) = t, \dots, TVal_I(\varphi_n) = t$.
- $TVal_I(\Gamma) = f$ de lo contrario.

Esto significa que un grupo de reglas se trata como una conjunción. El grupo vacío se trata como una tautología, así que $TVal_I(\text{Group}()) = t$.

5. Interpretación de documentos

Las fórmulas de documentos se interpretan utilizando las estructuras semánticas habituales con dos advertencias: `rif:local` debe renombrarse aparte y los documentos importados deben tenerse en cuenta.

Definición (cambio de nombre de constantes locales). Un *mapeo de renombrado*, ρ , es una función que mapea fórmulas de documentos a fórmulas de documentos, sujeto a la siguiente restricción:

- Si $\rho(\Delta) = \Delta'$ entonces Δ' es exactamente como Δ , excepto que todas las ocurrencias de algunas constantes `rif:local` en Δ pueden ser consistentemente renombradas en otras constantes `rif:local`. El cambio de nombre constante significa que a diferentes ocurrencias de la misma constante en Δ se renombraron de manera idéntica.

Definición (multi-estructura semántica). Una *multi-estructura semántica* \hat{I} es un par (\hat{I}_{ren}, I) , donde

- \hat{I}_{ren} es un mapeo de cambio de nombre; e
- I es una estructura semántica.

A continuación se define la semántica de documentos RIF:

Definición (Valor de verdad de las fórmulas Document). Sea Δ una fórmula Document y sean $\Delta_1, \dots, \Delta_n$ todas las fórmulas de documentos RIF-BLD que se importen en Δ (directa o indirectamente). Se usa $\Gamma, \Gamma_1, \dots, \Gamma_n$ para denotar las fórmulas Group respectivas asociadas con esos documentos. Si $\hat{I} = (\hat{I}_{ren}, I)$ es una multi-estructura semántica, entonces definimos:

- $TVal_{\hat{I}}(\Delta) = t$ si y solo si

$$TVal_I(\hat{I}_{ren}(\Gamma)) = TVal_I(\hat{I}_{ren}(\Gamma_1)) = \dots = TVal_I(\hat{I}_{ren}(\Gamma_n)) = t.$$

Es decir, Δ evalúa a t si y solo si, después del posible renombrado de constantes `rif:local`, su fórmula Group asociada, así como las fórmulas Group de todos los documentos importados evalúan a t .

Para fórmulas no documentales, se define $TVal_{\hat{I}}(\varphi) = TVal_I(\hat{I}_{ren}(\varphi))$.

Nótese que esta definición solo tiene en cuenta las fórmulas Document importadas a las que se puede acceder mediante las directivas Import de argumento único. También tenga en cuenta que algunos de los Γ_i anteriores pueden estar ausentes ya que todas las partes en una fórmula de documento son opcionales. En este caso, suponemos que Γ_i es una tautología, como `And()`, y cada función de $TVal$ asigna un Γ_i al valor de verdad t .

6. Implicación lógica

Ahora se define qué significa para un conjunto de reglas RIF-BLD (integradas en un grupo o en una fórmula Document) incluir otra fórmula RIF-BLD. En RIF-BLD se utiliza principalmente en la vinculación de fórmulas de condición, que pueden verse como consultas a grupos o documentos RIF-BLD. La implicación de las fórmulas de condición proporciona una base formal para las consultas RIF-BLD.

Definición (Modelos). Una multi-estructura \hat{I} es un *modelo* de una fórmula, φ , escrita como $\hat{I} \models \varphi$, si y solo si $TVal_{\hat{I}}(\varphi) = t$. Aquí φ puede ser una fórmula Documento o una fórmula no documental.

Definición (Implicación lógica). Sean φ y ψ fórmulas (Document o no). Decimos que φ *implica* ψ , escrito como $\varphi \models \psi$, si y solo si para cada multi-estructura, \hat{I} , $\hat{I} \models \varphi$ implica $\hat{I} \models \psi$.

Nótese que una consecuencia de la semántica de documentos múltiples de RIF-BLD es que las constantes locales especificadas en un documento no se pueden consultar desde otro documento.

Por ejemplo, si un documento, Δ' , tiene el hecho

```
"http://example.com/ppp"^^rif:iri("abc"^^rif:local)
```

Mientras que otra fórmula Document, Δ , importa a Δ' y tiene la regla

```
"http://example.com/qqq"^^rif:iri(?X):-
```

```
"http://example.com/ppp"^^rif:iri(?X),
```

```
Luego  $\Delta \models$  "http://example.com/qqq"^^rif:iri("abc"^^rif:local)
```

No se cumple.

Esto se debe a que el símbolo "abc"^^rif: local en Δ' y Δ es tratado como diferentes constantes mediante multi-estructuras semánticas.

Este comportamiento de los símbolos locales debe contrastarse con el comportamiento de los símbolos rif: iri. Supongamos que, en el escenario anterior, Δ' también tiene el hecho

```
"http://example.com/ppp"^^rif:iri("http://cde.example.org"^^rif:iri).
```

Entonces $\Delta \models$

```
"http://example.com/qqq"^^rif:iri("http://cde.example.org"^^rif:iri)
```

Se cumple.

Teniendo en cuenta esta semántica de RIF-BLD, ahora se establecen las sentencias de OWL que serán tenidas en cuenta para realizar la comparativa entre ambos formatos; se utilizan los axiomas de OWL, es decir, solamente las expresiones que enuncian una “operación aplicada sobre una entidad de una ontología (conceptos, relaciones, instancias, tipos de datos y valores); y todo axioma deberá ser representable en una estructura del tipo Grafárbol” (Chavarro, 2010).

Axiomas de OWL

El componente de una ontología OWL 2 que corresponde analizar en la presente propuesta, es un conjunto de axiomas: declaraciones que dicen lo que es verdadero en el dominio. OWL 2 proporciona un amplio conjunto de axiomas, que pueden ser declaraciones, axiomas sobre clases, axiomas sobre objetos o propiedades de datos, definiciones de tipos de datos, claves, aserciones (a veces también llamadas hechos) y axiomas sobre anotaciones (Motik, Patel-Schneider y Parsia, 2012).

Cada axioma tiene una de las siguientes formas:

1. Subproperty statement
2. Disjoint properties statement
3. Inverse property statement
4. Functional property statement
5. Symmetric property statement
6. Transitive property statement
7. Datatype definition
8. Has-key statement
9. Same-individual statement
10. Different-individuals statement
11. Subclass statement

Comparativa con OWL

En la tabla 2 se realiza la comparativa entre las expresiones en RIF-BLD y las expresiones de axiomas OWL en sintaxis funcional. Se observan algunas claves importantes en cuanto a expresiones que no pueden ser “traducidas” a OWL.

De la tabla 2 se pueden hacer las siguientes observaciones:

- Los campos vacíos en expresiones funcionales OWL representan una de dos formas: o expresiones simples RIF que no corresponden a axiomas OWL (y por tanto no son tenidas en cuenta para el objetivo presente de extender el grafárbol), o sentencias RIF-BLD que no corresponden a ninguna expresión en OWL 2, y por tanto, son el punto de partida para extensión del grafárbol ontológico.
- La expresión `Group` que se compara con `And` no tiene la misma utilidad en ambos lenguajes, ya que representa una conjunción de reglas en el dialecto RIF-BLD, mientras que en OWL es usada para conjunciones dentro de la definición de aserciones sobre clases o individuos de una ontología: no para grupos de reglas.

Expresión de presentación RIF-BLD	Expresión funcional OWL
Const	
Var	
And	ObjectIntersectionOf
Or	ObjectUnionOf
:-	
Exists	Some
Forall	All
=	EquivalentObjectProperties
#	ClassAssertion
##	SubclassOf
->	
External	
Import	Import
Prefix	
Base	
Group	And
Document	
List	
OpenList	

Tabla 2. Comparativa RIF-BLD y OWL en sentencias

Otros aspectos de comparación

	RIF-BLD	OWL
Negación	No	Si
Cuantificación variable abarca todo el dominio	Si. la misma variable se puede asignar a un individuo abstracto o a un valor de datos concreto	No
Las constantes (por ejemplo, IRIs) que denotan individuos se pueden escribir en lugar de un valor de datos, y viceversa	Si	No

Tabla 3. Comparativa de RIF-BLD y OWL en otros aspectos

Cuantificación variable abarca todo el dominio: En la semántica directa de OWL 2, los dominios para interpretar individuos respectivamente, literales (valores de datos), son disjuntos. La disyuntiva implica que los valores de dato no pueden ser miembros de una clase y que los individuos no pueden ser miembros de un tipo de dato. RIF no hace tales distinciones (De Bruijn y Welty, 2013).

Las constantes (por ejemplo, IRI) que denotan individuos se pueden escribir en lugar de un valor de datos: las constantes RIF (por ejemplo, IRIs) que denotan individuos se pueden escribir en lugar de un valor de datos, como el valor de una propiedad con valores de datos o en sentencias *Membership* de tipo de datos; de manera similar para constantes que denotan valores de datos.

Ejemplo

Considere el tipo de dato `xs:string` y una combinación RIF-OWL DL que consiste en el conjunto que contiene sólo una ontología OWL 2 DL en la que se encuentra:

```
ex:myiri rdf:type ex:A .
```

Y un documento RIF que contiene el siguiente hecho:

```
ex:myiri[rdf:type -> xs:string]
```

Esta combinación no se satisface directamente en RIF-OWL, porque `ex:myiri` es un identificador individual y `S` asigna identificadores individuales a elementos en `O`, que es disjunta de los elementos en el tipo de dato `xs:string` (De Bruijn y Welty, 2013).

Considere una combinación RIF-OWL DL que consiste en el conjunto que contiene sólo la ontología OWL 2 DL:

```
ex:hasChild rdf:type owl:ObjectProperty .
```

Y un documento RIF que contiene el siguiente hecho:

```
ex:myiri[ex:hasChild -> "John"]
```

Esta combinación no se satisface directamente en RIF-OWL, porque `ex:hasChild` es una propiedad, y los valores de las propiedades del objeto pueden no ser valores de datos concretos.

Considere una combinación RIF-OWL DL que consiste en la ontología:

```
SubClassof (ex:A ex:B)
```

Y un documento RIF que contiene la siguiente regla:

```
forall ?x (?x[rdf:type -> ex:A])
```

Esta combinación no se satisface directamente en RIF-OWL, es decir que no se puede aplicar en OWL, porque la regla requiere que cada elemento, incluido cada valor de datos concreto, sea miembro de la clase `ex:A`. Sin embargo, dado que cada interpretación de OWL requiere que cada miembro de `ex:A` sea un elemento del dominio del objeto, los valores de datos concretos no pueden ser miembros del dominio del objeto.

Después de revisar la comparativa entre sentencias RIF-BLD y OWL, se puede concluir lo siguiente:

- Se ha seleccionado adecuadamente el dialecto RIF-BLD, para los objetivos de la presente propuesta, el cual corresponde a reglas lógicas (contrario a RIF-PRD, en el cual se direccionan reglas de producción). Dentro de este ámbito, el dialecto Core está inmerso en RIF-BLD.
- Comparativa con RDF: Como la sintaxis completa OWL 2 es igual a la sintaxis RDF y la semántica basada en RDF OWL 2 es una extensión de la semántica RDF, la definición de compatibilidad completa RIF-OWL 2 es una extensión de la compatibilidad RIF-RDF (De Bruijn y Welty, 2013).

- La principal característica de RIF es la expresión de reglas; en esto radica la limitación de OWL y por ello se debe abordar el problema de la unión de ontologías con reglas.
- Los cuantificadores existencial y universal (*forall* y *exists*) cumplen una función diferente en RIF-BLD, y aquí encontramos otra limitación de OWL: son usados dentro de la *rule implication*, es decir, dentro de la definición de una regla.
- En RIF-BLD el cuantificador *forall* tiene dos utilidades:
 1. Regla universal: si ϕ es una implicación de regla (*rule implication*) y $\forall V_1, \dots, \forall V_n, n > 0$, son variables distintas, entonces $\text{forall } \forall V_1 \dots \forall V_n (\phi)$ es una fórmula, llamada regla universal. Se requiere que todas las variables *libres* en ϕ ocurran entre las variables $\forall V_1 \dots \forall V_n$ en la parte de cuantificación. Una ocurrencia de una variable $\forall V$ es *libre* en ϕ si no está dentro de una sub-fórmula de ϕ de la forma $\text{exists } \forall V (\psi)$ y ψ es una fórmula. Las reglas universales también se denominan Reglas RIF-BLD (Boley y Kifer, 2013).
 2. Hecho universal: si ϕ es una fórmula atómica y $\forall V_1, \dots, \forall V_n, n > 0$, son variables distintas, entonces $\text{forall } \forall V_1 \dots \forall V_n (\phi)$ es una fórmula, llamada un Hecho Universal, siempre que todas las variables libres en ϕ ocurran entre las variables $\forall V_1 \dots \forall V_n$. Los hechos universales a menudo se consideran reglas sin premisas.
- Las propiedades reflexiva, irreflexiva, simétrica, anti simétrica, funcional, funcional inversa, y transitiva que denotan argumentos para las relaciones (propiedades) entre clases e individuos en una ontología, no tienen correspondencia en RIF-BLD.

- La expresión característica principal de RIF-BLD es la *rule implication* (‘ : -‘), y dentro de ella se pueden agregar las demás sentencias RIF-BLD objeto de estudio (cuantificadores, grupos, entre otros). Por tanto, esta es la expresión que se debe extender en el lenguaje declarativo (especificación estructural del grafárbol), incluyendo sus diferentes variaciones. Las demás sentencias corresponden a declaraciones de variables, constantes, importación de documentos, las cuales no han sido tenidas en cuenta en (Chavarro, 2010) debido a que “El cambio se representa de manera independiente del lenguaje que se utiliza para implementar la ontología, mediante operadores declarativos. Un operador declarativo, en este contexto, es aquel que permite describir el requerimiento de cambio sin el detalle de cómo se debe implementar la operación que representa” lo que indica que los operadores del lenguaje declarativo corresponden con los axiomas o afirmaciones sobre los objetos.

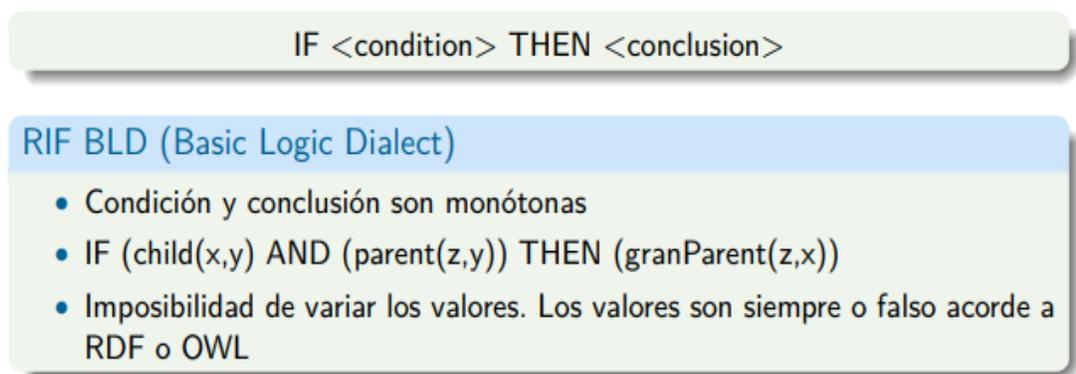


Figura 8. Rule implication en RIF-BLD. Basada en (Lera, 2014)

En el siguiente apartado se realizará la extensión del grafárbol ontológico, y por ende, del lenguaje declarativo, teniendo en cuenta la *rule implication*, descrita anteriormente, como representación principal de RIF-BLD y de la expresión de reglas en un formato estándar.

5.3. Extensión del grafárbol ontológico

En el presente apartado se realiza la extensión del grafárbol ontológico, teniendo en cuenta las limitaciones que éste presenta para la representación del conocimiento, de acuerdo con (Chavarro, 2010).

Reglas y Ontologías

El problema de integrar reglas y ontologías ha sido abordado desde diferentes perspectivas (Pomarolli, Anderlik y Küng, 2012; Carral Martínez y Hitzler, 2012; Horrocks, Patel-Schneider, Boley, Tabet, Grosz, y Dean, 2004; Espín, 2016 y Maeda, Ohta y Kuwabara, 2014).

Un sistema basado en reglas se usa para implementar un sistema inteligente como un sistema experto (Maeda, *et al.*, 2014). Las reglas ayudan en la limitación del poder expresivo de los lenguajes de ontologías, permiten un mayor grado de expresividad del que ofrece OWL. Cuando OWL no alcanza la expresividad necesaria, se utilizan las reglas, las cuales disponen de bases de razonamiento eficientes sobre ellas.

Como se menciona en (Carral Martínez y Hitzler, 2012), se define formalmente un conjunto de reglas lógicas de primer orden, C-Rules, que se pueden expresar dentro de OWL extendido con conjunción de roles. Así mismo, en (Pomarolli, *et al.*, 2012), se desarrolla un plugin para Protégè llamado R2RIF, el cual actualmente es compatible con los lenguajes basados en Lógica de Horn

(reglas / ontologías), pero los autores planean ampliarlo para proporcionar un mecanismo de importación genérico para admitir todo tipo de lenguajes de reglas.

En (Espín, 2016), se razona con SWRL combinando con una ontología sobre recomendaciones nutricionales, teniendo en cuenta que SWRL extiende los axiomas OWL con cláusulas de Horn aportando mayor expresividad y capacidad de razonamiento a la base de conocimiento.

Se observa un tema en discusión muy reciente, el cual abordaremos desde la perspectiva del modelado conceptual, extendiendo el grafárbol ontológico con la expresividad de las reglas, de acuerdo con RIF-BLD.

Definiciones

El modelo conceptual (Grafárbol), el lenguaje declarativo y la extensión a ODM constituyen el marco de referencia para la gestión conceptual del cambio ontológico (Chavarro, 2010). Esto quiere decir que para los objetivos de la presente propuesta, se busca adicionar a los operadores declarativos la funcionalidad de la implicación (rule implication), así como adicionar a la Ontología de Cambio Genérico (objetivo del siguiente apartado) las operaciones de cambio resultantes en el proceso. Por tanto, a continuación se define el concepto de Ontología y de Grafárbol ontológico dados dentro de este modelo conceptual:

Definición de ontología

Se utiliza la definición de ontología dada en (Euzenat y Shvaiko, 2007) usada también en (Chavarro, 2010):

Una ontología es una tupla $\mathcal{O} = (C, I, R, T, V, \leq, \perp, \epsilon, =)$, tal que

- C , es el conjunto de conceptos o clases, se interpretan como un conjunto de individuos y constituyen los elementos principales de una ontología.
- I , es el conjunto de individuos, objetos o instancias, interpretados como elementos particulares de un dominio.
- R , es el conjunto de relaciones, son un subconjunto del producto del dominio, y corresponden al concepto ideal de relación. Las entidades se pueden conectar por diversos tipos de relaciones, incluyendo: Especialización, Partición, o Relaciones Funcionales.
- T , es el conjunto de tipos de datos.
- V , es el conjunto de valores. (C, I, R, T, V siendo cada par disjuncto) son valores simples.
- \leq , es una relación definida sobre $(C \times C) \cup (R \times R) \cup (T \times T)$ llamada especialización.
- \perp , es una relación definida sobre $(C \times C) \cup (R \times R) \cup (T \times T)$ llamada exclusión.
- \in , es una relación definida sobre $(I \times C) \cup (V \times T)$ llamada instanciación.
- $=$, es una relación definida sobre $I \times T \times (I \cup V)$ llamada asignación.

Definición de Grafárbol ontológico

Un grafárbol ontológico G_{gfo} es una 6-tupla (Chavarro, 2010):

$$G_{gfo} = (N(G_{gfo}), A(G_{gfo}), \Psi_{gfo}, E(G_{gfo}), F_{G_{gfo}}, H_{gfo})$$

Donde:

$N(G_{gfo}) = \{n_i | n_i \in \text{TiposDeNodo}, i = 1, 2, 3, \dots, n\}$, $N(G_{gfo}) \neq \Phi$. Es decir, $N(G_{gfo})$ es el conjunto de nodos.

$A(G_{gfo}) = \{a_j | a_j \in \text{TiposDeArco}, j = 1, 2, 3, \dots, m\}$, con $N(G_{gfo}) \cap A(G_{gfo}) \neq \Phi$ disjuntos. Es decir $A(G_{gfo})$ es el conjunto de arcos.

Ψ_{gfo} : si $a_k \in A(G_{gfo})$ y $(n_i, n_j) \in N(G_{gfo})$, entonces, $\Psi(a_k) = (n_i, n_j)$; donde (n_i, n_j) , es un par ordenado, n_i nodo inicio y n_j nodo final.

$E(G_{gfo}) = \{e_j | e_j \text{ son etiquetas}, j = 1, 2, 3, \dots, s\}$. Las etiquetas o anotaciones $(e_0, e_1, e_2, \dots, e_s, \dots)$ corresponden al conjunto finito y disjunto en un lenguaje L, de nombres de etiquetas y de vértices. $E(G_{gfo}) \neq \Phi$

$F_{gfo}(n_i) = e_j$: cada par (n_i, e_j) es único y está determinado por una función de asociación entre el elemento i del conjunto de nodos y el elemento j del conjunto de etiquetas.

$H(a_m) = e_n$: establece una relación funcional única (a_m, e_n) , entre el elemento m del conjunto de arcos y el elemento n del conjunto de etiquetas.

Para representar una ontología en un grafárbol ontológico se debe tener en cuenta:

- Los nodos terminológicos corresponden a conceptos o tipos de datos.
- Los nodos individuo representan individuos nombrados o anónimos.
- Los arcos representan relaciones jerárquicas o funcionales entre nodos de cualquier tipo.

De RIF-BLD al marco de referencia Ontoconcept

Dentro de las fases de desarrollo en (Chavarro, 2010), se plantea:

1. Identificar las operaciones que se pueden aplicar a los conceptos, relaciones e instancias en los niveles de representación semántico y pragmático,
2. Representar estas operaciones en un modelo conformado por medio de estructuras, operadores y restricciones,
3. Extender el meta modelo de definición ontológica ODM, incorporando las operaciones de cambio, facilitando su utilización en los modelos construidos en el lenguaje de modelado UML, y
4. Construir un marco de referencia (framework) que permita implantar el modelo conceptual para la gestión del cambio ontológico.

A partir de las fases uno y dos, y dentro de la “definición de un conjunto de operadores declarativos (para la gestión del cambio ontológico), para realizar operaciones de los niveles semántico y pragmático” (Chavarro, 2010), descritos en la fase dos (y los cuales se resumen en las tablas 4, 5, 6, y 7), en la presente propuesta se plantea una extensión al grafárbol ontológico (que es un mecanismo conceptual para representar las diferentes entidades de una ontología) a través de la adición de operadores compuestos al lenguaje declarativo, que incorporen la *rule implication*, principal característica de RIF-BLD, y que tengan la misma representación en el grafárbol que sus operadores atómicos, de la misma manera que los operadores de la tabla 5.

Operador	Parámetros
Add/Delete/Update Concept	[name] [modificador] [disjoint, union disjoint] ExprC
Add/Delete/Update Relation	[name] [modificador] [opciones modificador] ExprR
Add/Delete/Update Data Type	[name] ExprTypeRange
Add/Delete/Update Individual	[name] [Is] ExprC

Tabla 4. Operadores conceptuales sobre entidades

Operador	Parámetros
Add/Delete Assertion	ExprC
Add/Delete Assertion Concept	[name] [modificador] ExprC
Add/Delete Assertion Relation	[name] [modificador] [ExprR] [ExprC] [List exprC]
Add/Delete Assertion Datatype	[name] [modificador] exprRange
Add/Delete Assertion Individual	[name] [Is] ExprC

Tabla 5. Operadores sobre aserciones o axiomas, basada en (Chavarro, 2010)

Operador	Parámetros
Add / Update ontology metadata	[modificadores]
Add / Update ontology annotation	[modificadores]

Tabla 6. Operaciones sobre metadatos y operaciones de anotación

Operador	Parámetros
Matching ontology	[lista de ontologías] [parámetros de técnicas a utilizar]
Merge ontology	[lista de ontologías] [alineación]
Integration ontology	[lista de ontologías] [alineación].

Tabla 7. Operaciones que involucran múltiples ontologías

Operadores propuestos

“Los operadores compuestos se pueden desagregar en múltiples operaciones atómicas y trabajan sobre una sola ontología. Constituyen una capa superior del conjunto de operadores atómicos, en tanto que se pueden describir en términos de operaciones atómicas” (Chavarro, 2010).

Lo anterior describe la forma en que los nuevos operadores sobre axiomas deben ser implementados de manera compuesta (y se requiere en la medida en que estos operadores agregados derivan en uno o más operadores que no son atómicos).

En la tabla 8 se especifican los operadores propuestos. Los tres operadores corresponden a las formas de Rule Implication que hay en la sintaxis del Basic Logic Dialect. Su representación en la estructura de tipo Grafárbol es tal como en una afirmación inferida: “Una afirmación inferida, es aquella que se representa mediante un camino entre dos nodos del Grafárbol ontológico. El camino es el conjunto formado por nodos y arcos que permite conectar dos nodos no adyacentes del Grafárbol ontológico.

Operador		Modificador	Parámetros y modificadores	Opciones Modificador
Add/Delete	Rule	ForAll	Add Assertion Relation	[name] [modificador] [parámetro modificador] <i>Implies</i> Add Assertion Relation [name] [modificador] [parámetro modificador]
Add/Delete	Rule	ForAll	[nameVariable]...[nameVariable]	(Add Assertion Relation [name] [parámetros y modificadores [nameVariable]] <i>And</i> Add Assertion Relation [nameVariable] [parámetros y modificadores]) <i>Implies</i> Add Assertion Relation [name] [parámetros y modificadores [nameVariable]]
Add/Delete	Rule	ForAll	[nameVariable]...[nameVariable]	(Add Assertion Relation [nameVariable] [parámetros y modificadores] <i>Some</i> [nameVariable] <i>Or</i> Add Assertion Relation [nameVariable] [parámetros y modificadores] <i>Some</i> [nameVariable]) <i>Implies</i> Add Assertion Relation [nameVariable] [parámetros y modificadores]

Tabla 8. Operaciones propuestas que involucran reglas

Ítem	Operador		Parámetro	Modificador	Parámetro Modificador
9	Add / Delete	Assertion Relation	[name]		
10	Add / Delete	Assertion Relation	[name]	Subrelation	exprR
11	Add / Delete	Assertion Relation	[name]	Max	exprR
12	Add / Delete	Assertion Relation	[name]	Min	exprR
13	Add / Delete	Assertion Relation	[name]	Exact	exprR
14	Add / Delete	Assertion Relation	[name]	All	exprR
15	Add / Delete	Assertion Relation	[name]	Only	exprR
16	Add / Delete	Assertion Relation	[name]	Some	exprR
17	Add / Delete	Assertion Relation	[name]	Not	exprC
18	Add / Delete	Assertion Relation	[name]	And	List exprC
19	Add / Delete	Assertion Relation	[name]	Or	List exprC
20	Add / Delete	Assertion Relation	[name]	Simmetric	exprR
21	Add / Delete	Assertion Relation	[name]	AntiSimmetric	exprR

22	Add / Delete	Assertion Relation	[name]	Transitive	exprR
23	Add / Delete	Assertion Relation	[name]	Reflexive	exprR
24	Add / Delete	Assertion Relation	[name]	Domain	exprRange
25	Add / Delete	Assertion Relation	[name]	Range	exprRange
26	Add / Delete	Assertion Relation	[name]	Composite Union	exprR
27	Add / Delete	Assertion Relation	[name]	Composite Disjoin	exprR
28	Add / Delete	Assertion Relation	[name]	Composite UnionDisjoin	exprR

Tabla 9. Operadores sobre axiomas (Relation). Basada en (Chavarro, 2010)

Estas reglas de representación, expresan la correspondencia que existe entre las afirmaciones expresadas en un lenguaje usado en la representación del conocimiento (KIF, OWL, RDF,...) y su representación en el Grafárbol” (Chavarro 2010).

En la tabla 8 todos los aspectos de parámetro y modificador de la **Assertion Relation** corresponden a la propuesta inicial del lenguaje declarativo, como se describe en la tabla 9. Es decir, este operador sobre axioma continúa empleándose en la misma forma, excepto en la utilización de la variable en los operadores propuestos dos y tres, la cual involucra el concepto o el individuo que se afecta con la regla **Rule**.

Los tres operadores son de naturaleza unaria (involucran una sola ontología), con soporte de trazabilidad, y afectan la teoría lógica de la ontología (son de afectación lógica). Presentan las mismas características de construcción y principios de diseño que los demás operadores del lenguaje declarativo (modularidad, extensibilidad, perspectiva de capas, reusabilidad y claridad en el nombrado), y se limitan a las mismas restricciones del conjunto de dichos operadores.

Se requiere un tipo diferente de repositorio para almacenar las *Rules* ya que “un Grafárbol ontológico (GAO) se caracteriza por: Expresividad: En un (GAO) se pueden representar 2 tipos de nodo (Términos e individuos), dos tipos de relación (jerárquicas y funcionales) y en forma implícita las restricciones del modelo” lo que quiere decir que la estructura del Grafárbol Ontológico no permite agregar las restricciones que aportarían los tres operadores propuestos, en forma de Nodos o de Arcos, o dicho de otra forma, estas operaciones no se definen en términos de su estructura. Dicho repositorio deberá ser leído y razonado una vez se agregue un operador que corresponda a la aserción que se implica en dichas reglas.

La implementación en el framework Ontoconcept de los operadores propuestos, a través de un plugin, no se ha considerado como objetivo de la presente investigación.

Resultados

Ejemplo 1.

El siguiente ejemplo se encuentra en (Morgenstern y Welty, 2013):

EN RIF:

```
Document (
  Prefix(rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
  Prefix(rdfs <http://www.w3.org/2000/01/rdf-schema#>)
  Prefix(imdbrel <http://example.com/imdbrelations#>)
  Prefix(dbpedia <http://dbpedia.org/ontology/>)
  Group (
    Forall ?Actor ?Film ?Role (
      If And(?Actor # imdbrel:Actor
            ?Film # imdbrel:Film
            ?Role # imdbrel:Character
            imdbrel:playsRole(?Actor ?Role)
            imdbrel:roleInFilm(?Role ?Film))
      Then dbpedia:starring(?Film ?Actor)
    )
  )
)
```

CONVERSIÓN AL LENGUAJE DECLARATIVO:

```
ADD RULE ForAll ActorV FilmV RoleV
  (ADD ASSERTION RELATION playsrole
    Domain ActorV
    Range RoleV
  And
    ADD ASSERTION RELATION roleinfilm
      Domain RoleV
      Range FilmV)
  Implies
    ADD ASSERTION RELATION starring
      Domain ActorV
```

Range FilmV

Aquí la regla procede cuando el objeto ActorV está presente en la relación Tiene un papel, y el papel está en una película; la regla define que se infiere la relación Actor protagoniza, para ese ActorV y esa FilmV.

Ejemplo 2.

El siguiente ejemplo se encuentra en (Morgenstern y Welty, 2013):

EN RIF:

```
Document (
  Prefix(rdfs <http://www.w3.org/2000/01/rdf-schema#>)
  Prefix(imdbrelf <http://example.com/fauximdbrelations#>)
  Prefix(dbpediaf <http://example.com/fauxdbpediarelations>)
  Prefix(ibdbrelf <http://example.com/fauxibdbrelations#>)
  Group(
    Forall ?Actor (
      If Or(Exists ?Film (imdbrelf:winAward(?Actor ?Film))
          Exists ?Play (ibdbrelf:winAward(?Actor ?Play)) )
      Then dbpediaf:awardWinner(?Actor)
    )
  )
)
```

CONVERSIÓN AL LENGUAJE DECLARATIVO:

```
ADD RULE ForAll ActorV (
  ADD ASSERTION RELATION WinAwardF
    Domain ActorV
    Range FilmV
    Some FilmV
  Or
  ADD ASSERTION RELATION winAwardP
```

```
Domain ActorV
Range PlayV
Some PlayV)
Implies
  ADD ASSERTION RELATION AwardWinner
  Domain ActorV
```

Aquí se tiene que distinguir entre las dos diferentes relaciones WinAward porque en RIF es indiferente la identificación de nombres similares para Películas y para Obras de teatro. La regla indica que un actor es Ganador por cualquiera de las dos opciones: si tiene un premio por una película o por una obra.

De los ejemplos descritos se concluye que no hay correspondencia semántica entre RIF y el lenguaje declarativo, o dicho de otra forma, entre RIF y OWL en las sentencias de axiomas (existen expresiones RIF-BLD que no corresponden a ninguna expresión en OWL 2). Se hace necesario la extensión de operadores en el lenguaje declarativo que permitan gestionar las reglas de RIF.

5.4. Extensión de la Ontología de Cambio Genérico

En este apartado, se comprende el concepto de Ontología de Cambio Genérico, a partir de la propuesta realizada en (Chavarro, 2010), de adición de las operaciones de cambio resultantes del lenguaje declarativo.

Como se especifica en Palma (2009), los enfoques anteriores para la representación de los cambios en una ontología son dependientes del modelo ontológico subyacente y consideran los cambios a nivel de entidad (conceptos, propiedades, individuos). Palma propone un modelo de capas para la representación de los cambios ontológicos que integra muchas de las características de las ontologías de cambio existentes.

Este modelo consiste en una Ontología De Cambio Genérico, independiente del modelo ontológico subyacente, que modela operaciones genéricas en una taxonomía de cambios que se espera sean compatibles con cualquier lenguaje de ontologías (ver figura 9).

En Chavarro (2010), se extiende dicho modelo con las operaciones de cambio que puedan provenir del lenguaje declarativo, estableciendo una correspondencia entre éste y la Ontología de Cambio Genérico, “como un mecanismo de correspondencia entre el lenguaje declarativo y cualquier lenguaje de implementación ontológica que posea su propia ontología de cambio”.

La figura 10 muestra el modelo propuesto en (Chavarro, 2010). Se observa la extensión de la ontología de cambio genérico con la clase *CambioComplejo* como respuesta a los operadores del lenguaje declarativo que involucran más de una ontología, es decir, aquellos operadores de mezcla e integración.

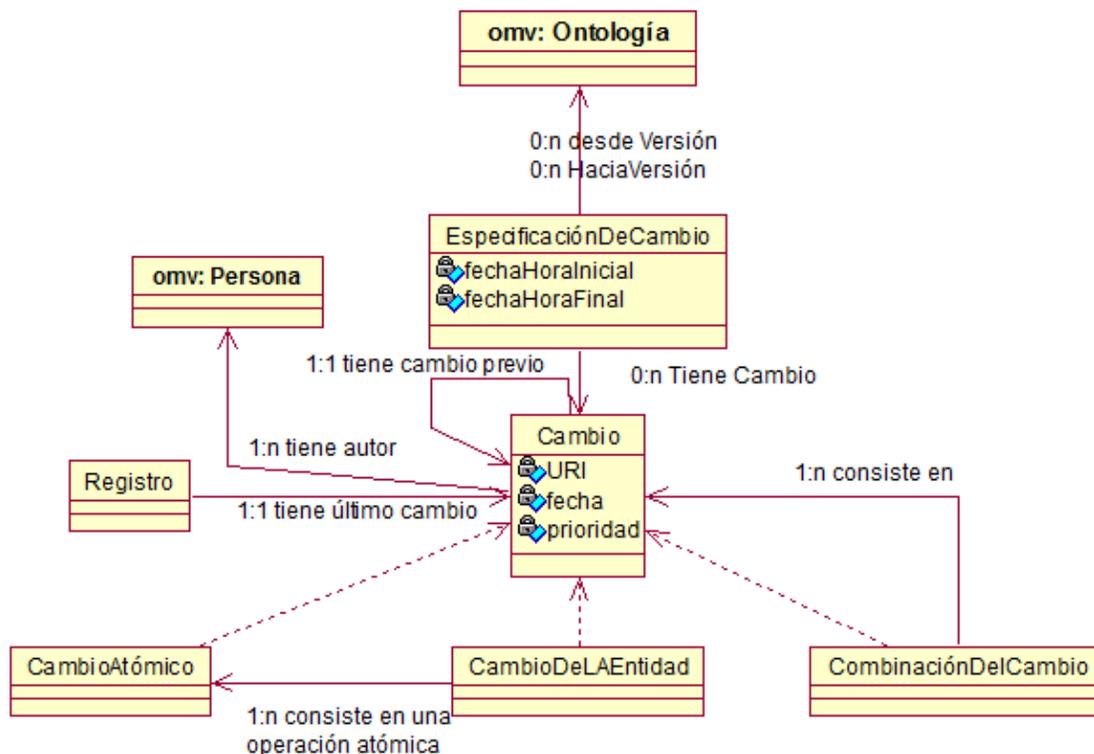


Figura 9. Ontología de Cambio Genérico, clases principales. Basado en (Palma, 2009).

Además, dentro de las operaciones atómicas se agregaron las de adición y borrado de conceptos, y a la clase *CambiodelaEntidad* se asociaron las operaciones ontológicas no atómicas que modifican conceptos y relaciones.

En la figura 11 se observa la propuesta abordada en la presente investigación, extendiendo la Ontología de Cambio Genérico con la clase Regla, que corresponde a un “Cambio Compuesto” (Es decir, una *CombinaciónDelCambio*).

Esta extensión se sustenta en el hecho que los cambios compuestos representan un grupo de cambios que, aplicados en conjunto, constituyen una entidad lógica (Palma, 2009). Ejemplos de estos cambios son: mover un subárbol de conceptos, adicionarlo, fusionar un conjunto de

conceptos hermanos, dividir un concepto. No es posible tener una lista completa de cambios compuestos, es decir, se pueden combinar cambios de entidad y cambios compuestos de muchas maneras diferentes.

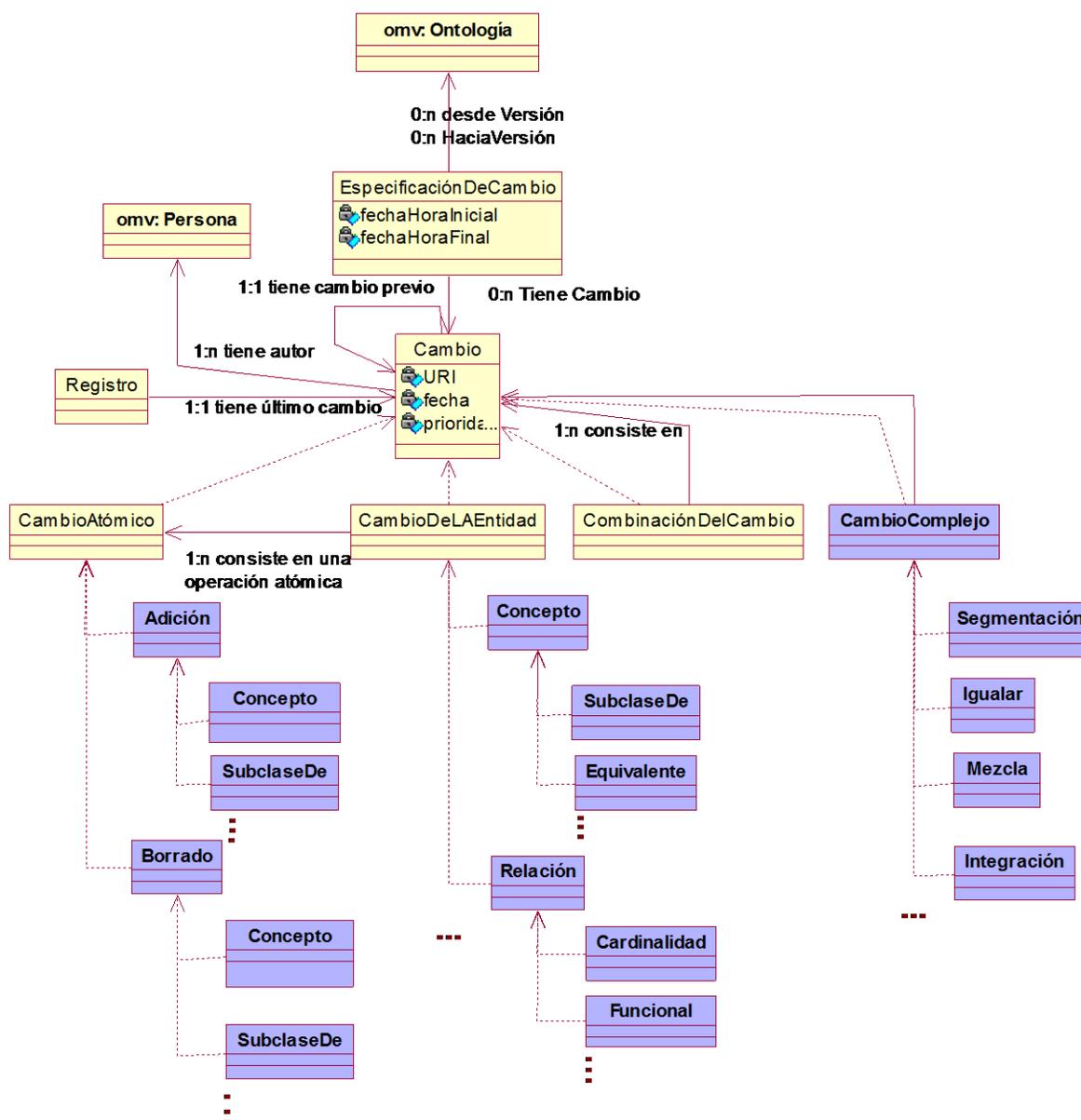


Figura 10. Extensión de la ontología de cambio genérico al lenguaje declarativo. Basado en Chavarro (2010)

Por ello, la ejecución de uno de los operadores propuestos para realizar una Rule Implication, se debe considerar un cambio compuesto ya que, además de modificar o adicionar relaciones entre conceptos, se ven en actuación varias operaciones atómicas.

Establecer la correspondencia entre el lenguaje declarativo, que contiene los operadores conceptuales de cambio ontológico, y la ontología de cambio genérico es un requisito para gestionar cambios en una ontología independientemente del lenguaje de implementación de la misma.

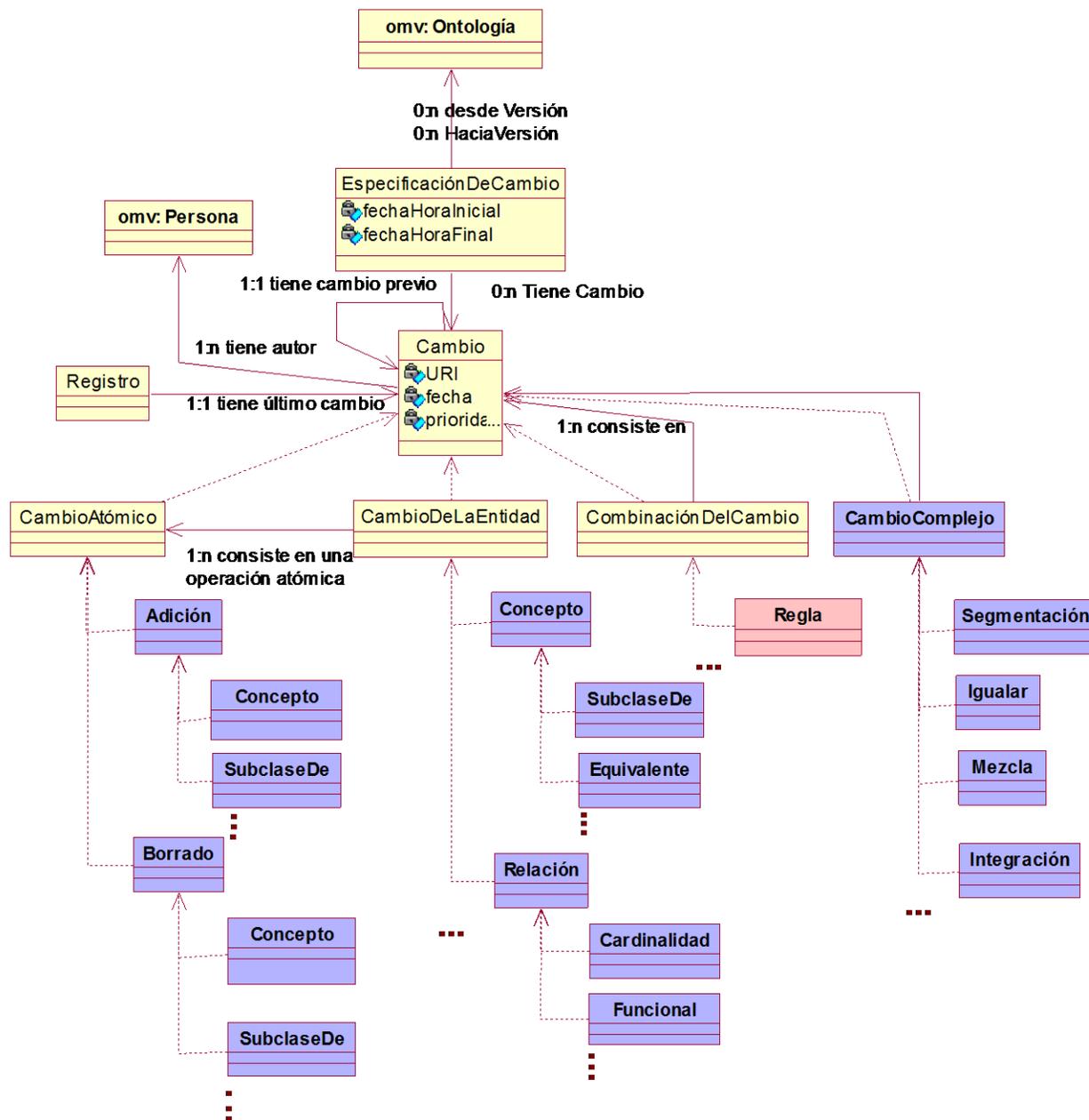


Figura 11. Extensión de la Ontología de Cambio Genérico al lenguaje declarativo con sentencias RIF-BLD.

Capítulo 6

Conclusiones

En el presente proyecto se ha optado por desarrollar una extensión para el lenguaje declarativo que tenga en cuenta el uso de RIF-BLD. La utilidad de RIF radica en el intercambio de reglas a través de la web; por ejemplo, puede proporcionar una representación neutral de los proveedores cuando se requiere una negociación de contratos de eBusiness en plataformas de reglas (Paschke, Morgenstern, Hirtle, Ginsberg, Patranjan, y McCabe, 2013). Esto permite a los desarrolladores del sistema de reglas y a las partes interesadas hacer su trabajo: por ejemplo, realizar inversiones en productos sin preocuparse por el bloqueo del proveedor, y sin preocuparse que un socio comercial no tenga la misma tecnología del proveedor.

El proyecto aporta las siguientes conclusiones:

- Se consigue establecer una comparativa de los elementos estructurales de RIF-BLD con las sentencias axiomáticas de OWL, encontrando mayor expresividad en RIF-BLD así como diferencias en cuanto a la expresión de la Negación, Group, entre otras.
- Las propiedades reflexiva, irreflexiva, simétrica, anti simétrica, funcional, funcional inversa, y transitiva que denotan argumentos para las relaciones (propiedades) entre clases e individuos en una ontología, no tienen correspondencia en RIF-BLD.
- La expresión característica principal de RIF-BLD es la *rule implication* (' : -'), y dentro de ella se pueden agregar las demás sentencias RIF-BLD objeto de estudio (cuantificadores, grupos, entre otros). Por tanto, esta es la expresión que extiende al lenguaje declarativo.

- De los ejemplos descritos se concluye que no hay correspondencia semántica entre RIF y el lenguaje declarativo (existen expresiones RIF-BLD que no corresponden a ninguna expresión en OWL 2). Se hace necesario la extensión de operadores en el lenguaje declarativo que permitan gestionar las reglas de RIF.
- Comparativa con RDF: Como la sintaxis completa OWL 2 es igual a la sintaxis RDF y la semántica basada en RDF OWL 2 es una extensión de la semántica RDF, la definición de compatibilidad completa RIF-OWL 2 es una extensión de la compatibilidad RIF-RDF.
- Es posible extender el lenguaje declarativo con tres operadores compuestos que representen las sentencias *rule implication*.
- Como se menciona también en (Chavarro, 2010), es muy útil una implementación de consultas RQL para buscar las relaciones (propiedades) existentes o almacenadas, y aplicar las reglas o adicionar unas nuevas.

Recomendaciones y trabajo futuro

Se pueden efectuar otras implementaciones de RIF teniendo en cuenta los siguientes aspectos:

- Hacer un desarrollo tecnológico con pruebas en Ontoconcept, ya que “la implementación de la herramienta ha seguido un curso iterativo e incremental, fundamentado en un framework para plugins en Java, que facilita incorporar de manera gradual los requerimientos tecnológicos” (Chavarro, 2010).

- Realizar la extensión del lenguaje declarativo con el dialecto de reglas de producción RIF-PRD.
- Hacer ampliaciones con otros ejemplos de ontologías acerca de aspectos que no fueron tenidos en cuenta en la presente propuesta.
- Igualmente continuar aplicando cada dialecto RIF que sea publicado posterior a la presente propuesta.

ANEXOS

Glosario y acrónimos

Cláusula de Horn: una cláusula con a lo sumo un literal positivo, es decir, innegado.

Cláusula definida: una cláusula Horn con exactamente un literal positivo.

Common Logic (CL): framework para la familia de lenguajes lógicos, basados en lógica de primer orden.

Datalog: Lenguaje de programación declarativo que sintácticamente es un subconjunto de Prolog.

Framework FLD: Es un formalismo para especifica todos los dialectos lógicos de RIF.

IRI: Identificadores de Recursos Internacionales. Al igual que sucede con las URL, las IRIs son un estándar propuesto por la W3C.

KIF: es un lenguaje lógico basado en lógica de primer orden que fue creado con el objetivo de actuar como interlingua entre diferentes formalismos y lenguajes de representación.

Negación no monótona: nuevos conocimientos pueden invalidar la premisa de negación que antes se consideraba una verdad.

ODM: Meta modelo de Definición Ontológica.

OMV: Ontología de vocabulario de metadatos (Ontology Metadata Vocabulary) para la web semántica.

OMG: Object Management Group.

RDF: Resource Description Framework (Marco de Descripción de Recursos).

RQL: (Resource Query Language) Un lenguaje de consulta diseñado para usar en URIs con estructuras de datos orientadas a objetos.

Sistema experto: sistema computacional que emula la capacidad de tomar decisiones de un humano experto.

SPARQL: (Protocol and RDF Query Language). Se trata de un lenguaje estandarizado para la consulta de triplas RDF.

SWRL: (Semantic Web Rule Language) Es un lenguaje propuesto para la Web Semántica que se puede usar para expresar reglas y lógica, combinando OWL DL u OWL Lite con un subconjunto de RDF.

Tautología: Una proposición compuesta es una tautología si es verdadera para todas las asignaciones de valores de verdad para sus proposiciones componentes.

UML: Lenguaje unificado de modelado.

XML: Extensible Markup Language

Referencias

- Álvarez, M. (2012). W3c Word wide web consortium. Huesca – VII Feria Virtual. Recuperado de <https://www.w3c.es/Presentaciones/2012/0417-WebSemantica-MA/0417-WebSemantica-MA.pdf>.
- Apache Software Foundation (2018). Hadoop MapReduce. Recuperado de <https://hadoop.apache.org>
- Berners-Lee, T. (2018). URIs Universal Syntax. World-Wide Web project. <https://www.w3.org/Addressing/URL/uri-spec.html>.
- Bertino, E., Provetti, A., y Salvetti, F. (2003). Local Closed-World Assumptions for reasoning about Semantic Web data. En: *APPIA-GULP-PRODE 2003, Joint Conference On Declarative Programming*. p 314–323.
- Boley, H. (2016). The RuleML Knowledge-Interoperation Hub. En: J.J. Alferes et al. (Ed.): *Rule Technologies. Research, Tools, and Applications 2016*, LNCS 9718, p 19–33. Springer.
- Boley, H., y Kifer, M. (2013). RIF Basic Logic Dialect (Second Edition). W3C Recommendation. Recuperado de <http://www.w3.org/TR/2013/REC-rif-bld-20130205/>.
- Boley, H., y Kifer, M. (2013b). RIF Framework for Logic Dialects (Second Edition). W3C Recommendation. 5 February 2013. Recuperado de <https://www.w3.org/TR/rif-fld/>.
- Boley, H., y Tabet, S., (2013). The Rule Markup Initiative. Recuperado de <http://ruleml.org/index.html>
- Brickley, D., y Guha, R.V. (2014). RDF Schema 1.1. W3C Recommendation. Recuperado de <http://www.w3.org/TR/rdf-schema/>.
- Carral Martínez, D., y Hitzler, P. (2012). Extending description logic rules. En: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. ESWC 2012. LNCS, vol. 7295, p 345–359. Springer, Heidelberg. Doi:10.1007/978-3-642-30284-8_30.

- Chavarro, J.C. (2010). Marco de referencia para la gestión del cambio en ontologías basados en Modelos conceptuales (Tesis doctoral). Universidad del Valle, Cali-Colombia.
- Chein, M., y Mugnier, M. (2009). Graph-based Knowledge Representation. Computational Foundations of Conceptual Graphs. Londres, UK: Springer.
- Chen, Ch., y Matthews, M. M. (2007). Extending Description Logic for Reasoning about Ontology Evolution. *IEEE/WIC/ACM International Conference on Web Intelligence*.
- Chen, Y., Xu, Z., Ni, Y., Cao, G., y Zhang, Sh. (2014). A RIF Based Mapping of RDB2RDF. En: *International Journal of Database Theory and Application*. Vol.7, No.6, p 29-44. Recuperado de <http://dx.doi.org/10.14257/ijdta.2014.7.6.04>.
- Colombo, G., Pugliese, D., Klein, P., y Lützenberg, J. (2014). A study for neutral format to exchange and reuse engineering knowledge in KBE applications. En: *Engineering, Technology and Innovation (ICE), International ICE Conference*. Recuperado de <http://ieeexplore.ieee.org/document/6871565/>.
- Cyganiak, R., y Wood, D. (2014). RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation. Recuperado de <http://www.w3.org/TR/rdf11-concepts/>.
- Davis, R., Shrobe, H., y Szolovits, P. (1993). What Is a Knowledge Representation? *AI Magazine*. AAAI. 14(1). 17-33. Recuperado de <http://www.aaai.org/ojs/index.php/aimagazine/article/view/1029/947>.
- De Bruijn, J., y Welty, C. (2013). RIF RDF and OWL Compatibility (Second Edition). *W3C Recommendation, 5 February 2013*. Recuperado de <http://www.w3.org/TR/2013/REC-rif-rdf-owl-20130205/>.
- Enderton, H. B. (2001). *A Mathematical Introduction to Logic, Second Edition*. Academic Press.
- Espín, V. (2016). Sistemas de recomendación semánticos para la compartición de conocimiento y la explotación de Tesauros: un enfoque práctico en el ámbito de los sistemas nutricionales (Tesis doctoral). Universidad de Granada. Departamento de lenguajes y sistemas informáticos. Recuperado de <https://hera.ugr.es/tesisugr/26128123.pdf>.

- Euzenat, J. y Shvaiko, P. (2007). *Ontology Matching*. Springer-Verlag. Berlin Heidelberg.
- Fillottrani, P. (2005). *Lenguajes de intercambio de reglas*. Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur. Recuperado de http://sedici.unlp.edu.ar/bitstream/handle/10915/20757/Documento_completo.pdf?sequence=1
- Gruber, T. (1993). *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. Technical Report KSL-93-04, Knowledge Systems Laboratory, Stanford University, CA.
- Gu, Y., Sun, M. (2017). *Research on Parallel Frequent Pattern Discovery Based on Ontology and Rules*. En: *4th International Conference on Information Science and Control Engineering (ICISCE)*. Recuperado de <http://ieeexplore.ieee.org/document/8110360/>.
- Gupta, V., Aggarwal, A., y Ghose, U. (2016). *Rule based Information Leveraging from Business Invoices. Exploiting the limited structure*. En: *International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT)*. Recuperado de <http://ieeexplore.ieee.org/document/7514613/>.
- Herman, I. (2007). *Arquitectura de la web semántica [figura]*. Recuperado de <https://www.w3.org/2007/Talks/0507-rif/>.
- Higueras, E. (2010). *RIF: Nuevo paso hacia la web semántica*. Recuperado de http://www.tendencias21.net/Nuevo-paso-hacia-la-Web-Semantica_a4583.html
- Hitzler, P., Krötzsch, M., y Parsia, B. (2012). *OWL 2 Web Ontology Language Primer (Second Edition)*. W3C Recommendation. Recuperado de <http://www.w3.org/TR/2012/REC-owl2-primer-20121211/>.
- Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosz, B., y Dean, M. (2004). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. W3C Member Submission. Recuperado de <http://www.w3.org/Submission/SWRL/>.
- Kalinichenko, L., Stupnikov, S., Vovchenko, A., y Kovalev, D. (2014). *Rule-Based Multi-dialect Infrastructure for Conceptual Problem Solving over Heterogeneous Distributed Information Resources*. En: B. Catania et al. (ed.), *New Trends in Databases and Information Systems, advances in Intelligent Systems and Computing*. Springer.

- Kifer, M. (2008). Rule Interchange Format: The Framework. Web Reasoning and Rule Systems. Lecture Notes in computer science. 5341. p 1-11.
- Kifer, M., y Boley, H. (2013). RIF Overview, 2nd Edition. W3C Working Group Note, 5 February 2013. Recuperado de <https://www.w3.org/TR/rif-overview/>
- Kim, J. (2015). Authorization Conflict Problems in Combining RIF Rules with RDF Data. En: *IEICE TRANSACTIONS on Information and Systems*. Vol.E98-D No.4 p 863-871. Recuperado de http://search.ieice.org/bin/summary.php?id=e98-d_4_863.
- Knowledge Based Systems, Inc. (2017). IDEFØ function modeling method. University Dr. College Station, Texas. Recuperado de http://www.idef.com/idefo-function_modeling_method/
- Kumar, S., y Kamath, M. (2015). Design and implementation of competent web crawler and indexer using web services. En: *Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference*. Ramanathapuram, India. DOI: 10.1109/ICACCCT.2014.7019393.
- Lee, CS., Wang, MH., Hsiao, YC., y Tsai, BH. (2017). Ontology-based GFML agent for patent technology requirement evaluation and recommendation. En: *Soft Comput. Methodologies and application*. Germany. Recuperado de <https://doi-org/10.1007/s00500-017-2859-1>.
- Lera, I., (2014). Desarrollo de Aplicaciones para la Web Semántica: Modelización, Programación, Inferencia y Herramientas. Recuperado de <http://acsic.uib.es/isaaclera/ws/presentationTecno-RIF.pdf>.
- Maeda, R., Ohta, N., y Kuwabara, K. (2014). MapReduce-Based Implementation of a Rule System. En: Badica A., Trawinski B., Nguyen N. (ed.) *Recent Developments in computational Collective Intelligence. Studies in Computational Intelligence*, vol 513. Springer.
- Maedche, A. (2002). Ontology learning for the semantic Web. Universidad de Karlsruhe. Kluwer Academic publishers.
- Morgenstern, L., y Welty, Ch. (2013). RIF Primer (Second Edition). W3C Recommendation. Recuperado de <http://www.w3.org/TR/2013/NOTE-rif-primer-20130205/>.

- Morán, D., Vaquero, L.M., y Galán, F. (2011). Elastically ruling the cloud: specifying application's behavior in federated clouds. *2011 IEEE 4th International Conference on Cloud Computing*. Recuperado de <http://ieeexplore.ieee.org/xpl/articleDetails.jsp>
- Motik, B., Patel-Schneider, P. F., y Parsia, B. (2012). OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax. *W3C Recommendation 11 December 2012*. Recuperado de <https://www.w3.org/TR/owl2-syntax/>.
- Nan, Sh., Van Gorp, P., Korsten, H., Kaymak, U., Vdovjak, R., Lu, X., y Duan, H. (2015). DCCSS: A meta-model for dynamic clinical checklist support systems. En: *Model-Driven Engineering and Software Development (MODELSWARD), 2015 3rd International Conference*. Angers, France. Recuperado de <http://ieeexplore.ieee.org/document/7323108/>.
- Nikkilä, R., Nash, E., Wiebensohn, J., Seilonen, I., y Koskinen, K. (2012). Spatial inference with an interchangeable rule format. *International journal of geographical information science*. p 1210-1226. Recuperado de <https://doi.org/10.1080/13658816.2012.750323>.
- Object Management Group. (2018). Business Process Model and Notation. Recuperado de <http://www.bpmn.org/>.
- Object Management Group. (2014). ODM Ontology Definition Metamodel. Specification Version 1.1. <http://www.omg.org/spec/ODM/>
- Open Geospatial Consortium. (2018). OGC Simple Feature Access. Recuperado de <http://www.opengeospatial.org/>.
- Ouaret, Z., Boussaid, O., y Chalal, R. (2014). Towards the Automation of XML Data Warehouse Logical Design. En: *Digital Information Management (ICDIM), 2014 Ninth International Conference*. Phitsanulok, Thailand. DOI: 10.1109/ICDIM.2014.6991422.
- Palma de León, R. A. (2009). Ontology Metadata Management in Distributed Environments. Ph.D. thesis. Universidad Politécnica de Madrid, Facultad de Informática, Departamento de Inteligencia Artificial.
- Paschke, A., Morgenstern, L., Hirtle, D., Ginsberg, A., Patranjan, P.L., y McCabe, F. (2013). RIF Use Cases and Requirements (Second Edition). *W3C Recommendation*.

- 5 February 2013. Recuperado de <https://www.w3.org/TR/2013/NOTE-rif-ucr-20130205/>.
- Polleres, A., Boley, H., y Kifer, M. (2013). RIF Datatypes and Built-Ins 1.0 (Second Edition). W3C Recommendation 5 February 2013. Recuperado de <http://www.w3.org/TR/2013/REC-rif-dtb-20130205/>.
- Pomarolli, A., Anderlik, S., y Küng, J. (2012). R2RIF - Rule Integration Plugin for Protégé OWL. En: Moreno-Díaz R., Pichler F., Quesada-Arencibia A. *Computer Aided Systems Theory – EUROCAST 2011. Lecture Notes in Computer Science*, vol 6927. Springer.
- Rattanasawad, T., Runapongsa Saikaew, K., Buranarach, M., y Supnithi, T. (2013). A review and comparison of rule languages and rule-based inference engines for the Semantic Web. En: *International Computer Science and Engineering Conference (ICSEC): ICSEC 2013*. Recuperado de <http://ieeexplore.ieee.org/>.
- Sainte Marie, C. D., y Hallmark, G. (2013). RIF Production Rule Dialect (Second Edition). W3C Recommendation. Recuperado de <http://www.w3.org/TR/2013/REC-rif-prd-20130205/>.
- Skvortsov, N., Vovchenko, A., Kalinichenko, L., Kovalev, D., y Stupnikov, S. (2014). A metadata model for semantic search of rule-based workflow implementations. *Sistemy i Sredstva Inform.*, Volume 24, Issue 4, p 4–28. Recuperado de <http://dx.doi.org/10.14357/08696527140401>.
- Steele Jr., G.L. (1990). *Common LISP: The Language*. Second Edition, Digital Press.
- Stojanovic, L. (2004). *Methods and Tools for Ontology Evolution*. PhD thesis, University of Karlsruhe (TH), Germany.
- Veiga, F. (2011). *Implementation of the RIF-PRD (Tesis de Maestría)*. Universidad Nueva de Lisboa, Lisboa-Portugal. Recuperado de http://run.unl.pt/bitstream/10362/6310/1/Veiga_2011.pdf
- Verborgh, R., y De Roo, J. (2015). Drawing Conclusions from Linked Data on the Web: The EYE Reasoner. En: *IEEE Software*, Volume 32, Issue 3, May-June 2015. p 23 - 27. DOI: 10.1109/MS.2015.63.

- Wang, X., Chen, J., Cao, L., y Meng, X. (2013). The Foundation of Fuzzy Rule Interchange in the Semantic Web. En: *IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT)*. Recuperado de <http://ieeexplore.ieee.org/>.
- Wang, X., Meng, X., Chen, J., y Wang, B. (2013). Fuzzy Rule Interchange between f-R2ML and f-RIF. En: *10th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. p 600-604. Recuperado de <http://ieeexplore.ieee.org/>
- World Wide Web consortium. (2003). Extensible Markup Language. Recuperado de <http://www.w3.org/XML/Core/>.
- Zambon, E., y Guizzardi, G. (2017) Formal Definition of a General Ontology Pattern Language using a Graph Grammar. En: *Proceedings of the Federated Conference on Computer Science and Information Systems*. p 1–10. DOI: 10.15439/2017F001.