Un Repositorio RDF para la Integración de Flujos de Datos de Analítica Web en Comercio Electrónico

Maria del Mar Roldán-García, Jose García-Nieto, and Jose F. Aldana-Montes

Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga
{mmar, jnieto, jfam}@lcc.uma.es

Resumen La Analítica Web supone hoy en día una tarea ineludible para las empresas de comercio electrónico, ya que les permite analizar el comportamiento de sus clientes. El proyecto Europeo SME-Ecompass tiene como objetivo desarrollar herramientas avanzadas de análitica web accesibles para las PYMES. Con esta motivación, proponemos un servicio de integración de datos basado en ontologías para recopilar, integrar y almacenar información de traza web procedente de distintas fuentes. Estas se consolidan en un repositorio RDF diseñado para proporcionar semántica común a los datos de análisis y dar servicio homogéneo a algoritmos de Minería de Datos. El servicio propuesto se ha validado mediante traza digital real (Google Analitics y Piwik) de 15 tiendas virtuales de diferentes sectores y países europeos (UK, España, Grecia y Alemania) durante varios meses de actividad.

Keywords: Análisis Web, Ontologías, Integración de datos, RDF y Google Analytics.

1. Introducción

El Análisis Web está tomando cada vez más relevancia dentro del comercio electrónico, ya que permite a los comerciantes y gestores de sitios web obtener información relevante sobre el comportamiento de los clientes cuando visitan sus tiendas online. Además, las aplicaciones de analítica web ofrecen una visión cualitativa y cuantitativa sobre el posicionamiento de una web en el mercado y ayudan a obtener indicios sobre el impacto de cierta oferta o campaña publicitaria. Los procesos de análisis web se basan en el trazado de la "huella electrómica o digital" del visitante del sitio online, lo cual incluye desde metadatos de geolocalización, preferencias e incluso ratio de conversión, es decir, proporción de visitantes que terminan efectuando una compra. Estos datos se contrastan con indicadores de rendimiento (Key Performance Indicators, KPIs) para mejorar la tasa de éxito global del sitio de comercio electrónico analizado.

Actualmente existen un gran número de herramientas de analítica web, algunas de ellas muy conocidas, como: Google Analytics, Piwik y Clicky. Estas

herramientas se utilizan, no solo para trazar y medir el tráfico web en un sitio específico, si no también para analizar su actividad comercial, en términos de compras, beneficios, etc. Sin embargo, estas herramientas normalmente se centran en atributos numéricos de traza (contadores) y medidas de bajo nivel, sin la posibilidad de obtener análisis más sofisticados, como la clasificación de tipología de clientes y su evolución respecto a cierto tipo de producto. En la mayoría de los casos, estos análisis están disponibles sólo para sus versiones comerciales, las cuales son raramente accesibles para la pequeña y mediana empresa de comercio electrónico.

En este contexto, el proyecto europeo SME E-Compass [1] nace con el objetivo de generar herramientas de analítica web avanzadas y accesibles para las PYMES europeas. Estas aplicaciones software se nutren de diferentes fuentes de datos de traza web provenientes de huella electrónica (en forma de código *JavaSript*) integrada en las propias tiendas online. Sin embargo, la integración de información proveniente de múltiples fuentes de datos supone el tratamiento de modelos de datos diferentes, con esquemas y lenguajes de consulta heterogéneos.

Con esta motivación, proponemos en este trabajo un modelo semántico guiado por ontología para la recolección y el fusionado de datos de manera coherente, provenientes de traza web de servicios comerciales de huella electrónica. Como consecuencia, los datos procesados son anotados semánticamente y almacenados para su posterior utilización en el entrenamiento de algoritmos de minería de datos que analizan el comportamiento de los visitantes en sitios reales de comercio electrónico.

Nuestro modelo semántico utiliza una ontología de dominio como esquema de mediación, para la representación y consolidación de la semántica particular de los datos de traza web. Para la correspondencia entre los esquemas particulares de cada fuente de datos y nuestra ontología, hemos desarrollado una serie de funciones de *"mapeo (mappings)"* mediante las que se transforman los datos originales al formato estándar RDF (Resource Description Framework) [2]. De esta forma, todos los flujos de datos heterogéneos se almacenan en un repositorio RDF, sobre el cual se ofrende un servicio unificado de consulta para los algoritmos de análisis de alto nivel.

Por tanto, como aportación principal y original de este trabajo, hemos diseñado e implementado por primera vez una ontología OWL (Web Ontology Language) [1,4] para analítica web. Esta ontología contempla un conjunto de atributos y métricas de traza web, lo suficientemente exhaustivo y complementario, provenientes de las herramientas más representativas y utilizadas hoy en día para el análisis web: Google Analytics y Piwik.

Como aportación práctica, hemos validado nuestro modelo semántico mediante la captura e integración automática de diferentes flujos de datos de huella electrónica (Google Analytics y Piwik) alojada en 15 tiendas online reales de diferentes sectores comerciales (moda, belleza, turismo, electrónica, farmacia, gastronomía y venta al por menor en general) y países (Reino Unido, Grecia,

[1] SME-Ecompass FP7 European initiative http://www.sme-ecompass.eu/
[2] RDF in W3C https://www.w3.org/RDF/

Alemania y España), durante varios meses de actividad. Estos datos son por tanto integrados bajo un formato común y almacenados en un único repositorio RDF. Realizamos además varios casos de uso de aplicación, utilizando los datos integrados y anotados semánticamente, para el entrenamiento de algoritmos de minería de datos avanzados para el análisis del visitante web. En concreto, mediante estos algoritmos realizamos el análisis del comportamiento del visitante y su preferencia respecto a varios productos de una determinada marca comercial.

Este trabajo se organiza de la siguiente manera. En la siguiente Sección se presenta el modelo semántico propuesto, dando detalles de nuestra ontología, las fuentes de datos y el repositorio RDF desarrollado. En la Sección 3, se describe el caso de uso realizado para la validación del modelo semántico. Finalmente, la Sección 4 contiene las conclusiones y el trabajo futuro.

2. Modelo Semántico

El principal objetivo de este trabajo es recopilar, limpiar, consolidar e integrar información de diferentes fuentes de huella electrónica. Para ello se ha diseñado un modelo semántico cuyo elemento principal es una ontología que representa el conocimiento común del dominio de aplicación. En concreto, hemos utilizado la metodología "Ontology 101 development process"[5] para definir una ontología OWL que describe las principales características de las tiendas virtuales en siete pasos:

1. *Determinar el dominio y el ámbito de la ontología.* Inicialmente se tuvieron en cuenta las posibles variables de Google Analytics y de Piwik, además de las de los competidores de la tienda virtual, que son necesarias para los algoritmos de minería de datos. Por ejemplo: el origen y los atributo de los visitantes, los detalles de los productos y de los clientes, etc.
2. *Considerar la reutilización de ontologías ya existentes.* No hemos encontrado ontologías similares que se hayan desarrollado previamente para el análisis de datos de huella digital en comercio electrónico. Sin embargo, hemos tenido en cuenta parcialmente dos ontologías relacionadas: *GoodRelations* [3], que define un vocabulario estándar para comercio electrónico y la *Product Ontology*, que categoriza el tipo de producto basándose en Wikipedia.
3. *Enumerar los términos importantes en la ontología.* Los términos más importantes en la ontología se extrajeron en una fase previa de especificación de requisitos [2] a partir del conjunto mínimo de variables que se necesitan. Ejemplos de estos términos son: *Address*, *Visitor*, *Customer*, *Device*, *Browser*, *Geographical_origin*, *Number_of_visitors*, *Conversion_rate*, etc.
4. *Definir las clases y la jerarquía de clases.* A partir de la lista de términos importantes, obtenemos las clases de la ontología. La Figura 1 muestra el primer nivel de la jerarquía de clases, partiendo de la clase *Thing* ($\top$). Estas clases se relacionan con otras y algunas de ellas tienen subclases. Por ejemplo, *Bounce_rate*, *Total_revenue*, *Number_of_returning_visitors*, y *Number_of_transactions* son subclases de la clase *Analytic_parameters*.

5. *Definir las relaciones entre clases y sus atributos.* Para identificar las relaciones entre las clases (*object properties*) y los atributos de las clases (*data properties*) hemos tenido en cuenta el conjunto inicial de variables identificadas en el paso 1. Ejemplos de relaciones entre clases son: *an e-shop owner is owner of an e-shop*, *a visitor makes visits*, *a device has a browser*, *an IP address belongs to an organization*, etc. Ejemplos de atributos de clase son: *title and URL de una page*, *first and last name de un e-shop owner*, *version del operating system*, *duration de una visit*, etc. Para las subclases de *Analytic_parameters* se ha definido una relación para establecer la clase dominio de la misma. Por ejemplo, *Page* se relaciona con *Bounce_rate* y *Date_of_last_visit*; *E-shop* se relaciona con *Number_of_customers*. Los cuadros 2, 3, 5, and 6, describen un conjunto de las relaciones y atributos de las principales clases de la ontología.
6. *Definir las propiedades de los atributos.* Definición de las restricciones de cardinalidad y de valor (*value restrictions*). En nuestra ontología, las restricciones de valor se usan para especificar el tipo de datos válido en cada una de las *data properties* definidas para las subclases de *Analytic_parameters*. Por ejemplo, el rango de la propiedad *hasValue* se restringe a *float*, cuando su dominio es la clase *Bounce_rate*; el rango de la propiedad *hasValue* se restringe a *date*, cuando su dominio es la clase *Date_of_last_visit*.
7. *Crear las instancias (individuos).* Las instancias (individuos en OWL) corresponden a los datos de huella digital que se obtienen de Google Analytics, de Piwik, o del módulo de *srapping* de los competidores, para cada una de las tiendas virtuales. Estos datos se mapean a RDF teniendo en cuenta la ontología. También se han definido individuos para determinar los elementos específicos con los que se pueden relacionar algunas clases. Por ejemplo, cuando el dominio de la propiedad *hasType* es la clase *Article_number* su rango se restringe a los valores: "ASIN", "EAN", o "ISBN". "ASIN", "EAN", e "ISBN"se definen por tanto como instancias de la ontología.

2.1. Ontología

Como resultado del desarrollo anterior, nuestra ontología contiene un total de 62 clases (grupos de individuos que intercambian los mismos atributos), 61 propiedades de objeto (relaciones binarias entre los individuos), 33 restricciones de axioma y 3 individuos. La ontología completa, a la cual llamamos "wao.owl" (Web Analytics Ontology), puede consultarse a través de su enlace público en WebProtégé [3].

Por simplicidad, nos restringimos en este trabajo a la descripción de un subconjunto de las clases principales, incluyendo algunas de sus propiedades de objeto y de datos más interesantes y representativas. Estas clases son: *Analytics_parameters*, *E-shop*, *Visitor*, *Page* e *Item*. Cada una de estas clases requiere un conjunto de propiedades o condiciones para su contextualización, es decir, los individuos que satisfacen estas propiedades son miembros de estas clases.

[3] URL `http://stanford.io/1XHhHzr`

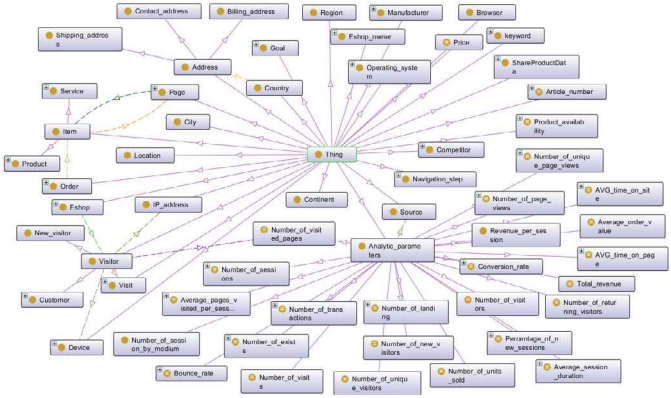

Figura 1. Vista general de la ontología WAO. Las flechas continuas se refieren a *subclase de(owl:subclassOf.* Las flechas discontinuas indican propiedades específicas.

- ***Analytics_parameters***. Atributos dependientes del tiempo que proveen Google Analytics y Piwik. Cara uno de estos parámetros tiene un valor (*hasValue* en Cuadro 1) que corresponde con el dato que provee la herramienta de analítica web. Además tiene una fecha (*hasDate*), que corresponde con el día y la hora en que se genera el dato. Las principales subclases de los parámetros de análisis ($\sqsubseteq$ Analytic_parameters) son, entre otras: *Average_order_value*, *Average_pages_visited_per_session*, *Average_session_duration*, *Average_time_on_site*, *Bounce_rate*, *Conversion_rate*, *Number_of_transactions*, *Number_of_landings*, *Number_of_new_visitors*, *Number_of_page_views*, *Revenue_per_ session* y *Total_revenue.* El Cuadro 1 contiene algunas propiedades de objeto y de datos más representativas de *Analytics_parameters*, perteneciendo cada parámetro analítico a un tipo de datos. Por ejemplo, el tipo referente al número de transacciones en la tienda online (*Number_of_transactions*) es un número entero no negativo y el valor del ratio de conversión (*Conversion_rate*) es un número real. Las restricciones de tipo de datos se incluyen mediante las propiedades de datos.

- ***E-shop.*** Una *E-shop* o tienda online tiene varias páginas (Pages) y un propietario (e-shop's owner). La e-shop tiene atributos como la latitud, la longitud y la zona horaria. El propietario puede tener competidores, que son a su vez propietarios de otras e-shops. Los parámetros de análisis de una e-shop son: *average_order_value*, *average_pages_visited_per_ session*, *average_ session_duration*, *average_time_on_site*, *conversion_rate*, *date_of_last_transaction*, *number_of_customers*, *number_of_failed_transactions*, *number_of_successful_transactions*, *number_of_ new_customers*, *number_of_new_visitors*, *number_of_sessions_by_medium*, *num-*

Cuadro 1. Grupo Analytics_parameters: las propiedades de objeto y de datos se representan en lógica de descripciones

Propiedades de Objeto	Lógica de Descripciones
hasBrowser	$\exists$ hasBrowser.Thing $\sqsubseteq$ Analytic_parameters $\sqcup$ Device $\top \sqsubseteq \forall$ hasBrowser.Browser
hasCity	$\exists$ hasCity.Thing$\sqsubseteq$ Analytic_parameters $\sqcup$ Location $\sqcup$ Visitor $\top \sqsubseteq \forall$ hasCity.City
hasRegion	$\exists$ hasRegion.Thing $\sqsubseteq$ Analytic_parameters $\sqcup$ Location $\sqcup$ Visitor $\top \sqsubseteq \forall$ hasRegion.Region
hasCountry	$\exists$ hasCountry.Thing $\sqsubseteq$ Analytic_parameters $\sqcup$ Location $\sqcup$ Visitor $\top \sqsubseteq \forall$ hasCountry.Country
hasContinent	$\exists$ hasContinent.Thing $\sqsubseteq$ Analytic_parameters $\sqcup$ Location $\top \sqsubseteq \forall$ hasContinent.Continent
hasSource	$\exists$ hasSource.Thing $\sqsubseteq$ Analytic_parameters $\top \sqsubseteq \forall$ hasSource.Source
Propiedades de Datos	**Lógica de Descripciones**
hasDate	$\exists$ hasDate.DatatypeLiteral $\sqsubseteq$ Analytic_parameters $\sqcup$ Price $\sqcup$ Product_availability $\top \sqsubseteq \forall$ hasDate.DatatypedateTimeStamp
hasHour	$\exists$ hasHour.DatatypeLiteral $\sqsubseteq$ Analytic_parameters $\top \sqsubseteq \forall$ hasHour.Datatypetime
hasNetworkDomain	$\exists$ hasNetworkDomain.DatatypeLiteral $\sqsubseteq$ Analytic_parameters $\top \sqsubseteq \forall$ hasNetworkDomain.Datatypestring
hasValue	$\exists$ hasValue.DatatypeLiteral $\sqsubseteq$ Analytic_parameters $\sqcup$ Article_number $\sqcup$ Price $\sqcup$ Product_availability

ber_of_transactions, number_of_unique_visitors, number_of_units_sold, number_of_visitors, number_of_visits, percentage_of_new_sessions, revenue_per_session, total_revenue y *number_of_returning_visitors.* Todos estos parámetros relacionados con la tienda online son dependientes de la dimensión temporal. Por tanto, estos parámetros se modelan como clases que se relacionan con la e-shop mediante las correspondientes propiedades de objeto. En el Cuadro 2 se recoge un subconjunto de propiedades de objeto y de datos con clases del grupo E-shop como dominio.

- ***Visitor/Visit***. La clase visitante (*Visitor*) tiene dos subclases: *Customer*, referente al cliente y *New_visitor*, que anota al nuevo visitante en la tienda online. Un cliente es un visitante que compra. Los clientes tienen nombre y dirección postal (provenientes del registro en la web), mientras que los visitantes no. Un visitante entra en la tienda online a través de un dispositivo (*Device*). Los parámetros de análisis para los visitantes son: *bounced_rate, number_of_visits* y *number_of_visited_pages*, mientras que el cliente tiene además el parámetro *number_of_transactions.*

Los visitantes acceden a páginas (*Pages*), por lo que efectúan visitas. Estas visitas son fundamentales para capturar el comportamiento del visitante. Cada visita tiene una página de entrada y otra de salida (que puede ser la misma). Además, también tiene una página de referencia o de enlace, a partir de la cual el visitante entra en la web: buscadores, redes sociales, webs con enlaces de publicidad, etc. Si esta página de referencia es un buscador, se pueden obtener las palabras clave utilizadas para realizar la búsqueda. Las visitas también tienen una duración, cuyos atributos dependen de la marca temporal desde que se accede a la página de entrada hasta que se sale por la página de salida; un indicador de si se ha realizado compra y el número total de artículos vendidos;

Cuadro 2. Grupo Eshop: las propiedades de objeto y de dato se representan en lógica de descripciones

Propiedades de Objeto	Lógica de Descripciones
hasVisitor	$> \equiv$ makesVisit$>^{-}$
	$\exists$ hasVisitor Thing $\sqsubseteq$ Eshop
	$\top \sqsubseteq \forall$ hasVisitor Visitor
hasNumberOfVisitors	$\exists$ hasNumberOfVisitors Thing $\sqsubseteq$ Eshop $\sqcup$ Page
	$\top \sqsubseteq \forall$ hasNumberOfVisitors Number_of_visitors
hasNumberOfVisits	$\exists$ hasNumberOfVisits Thing $\sqsubseteq$ Eshop $\sqcup$ Page $\sqcup$ Visitor
	$\top \sqsubseteq \forall$ hasNumberOfVisits Number_of_visits
isOwnerOf	$\exists$ isOwnerOf Thing $\sqsubseteq$ Eshop_owner
	$\top \sqsubseteq \forall$ isOwnerOf Eshop
Propiedades de Datos	Lógica de Descripciones
hasName	$\exists$ hasName DatatypeLiteral $\sqsubseteq$ Browser $\sqcup$ Competitor $\sqcup$ Eshop $\sqcup$ Goal $\sqcup$ Item $\sqcup$ Operating_system $\sqcup$ Page $\sqcup$ Product
	$\top \sqsubseteq \forall$ hasName Datatypestring
hasURL	$\exists$ hasURL DatatypeLiteral $\sqsubseteq$ Competitor $\sqcup$ Eshop $\sqcup$ Page $\sqcup$ Price
	$\top \sqsubseteq \forall$ hasURL Datatypestring

Cuadro 3. Grupo Visitor: las propiedades de objeto y de datos se representan en lógica de descripciones

Propiedades de Objeto	Lógica de Descripciones
hasDevice	$\exists$ hasDevice Thing $\sqsubseteq$ Visitor
	$\top \sqsubseteq \forall$ hasDevice Device
hasNumberOfVisits	$\exists$ hasNumberOfVisits Thing $\sqsubseteq$ Eshop $\sqcup$ Page $\sqcup$ Visitor
	$\top \sqsubseteq \forall$ hasNumberOfVisits Number_of_visits
hasCity	$\exists$ hasCity.Thing$\sqsubseteq$ Analytic_parameters $\sqcup$ Location $\sqcup$ Visitor
	$\top \sqsubseteq \forall$ hasCity.City
makesVisit	hasVisitor¿ $\equiv$ makesVisit¿$^{-}$
	$\exists$ makesVisit Thing $\sqsubseteq$ Visitor
	$\top \sqsubseteq \forall$ makesVisit Visit
Propiedades de Datos	Lógica de Descripciones
hasDaysSinceFirstVisit	$\exists$ hasDaysSinceFirstVisit DatatypeLiteral $\sqsubseteq$ Visitor
	$\top \sqsubseteq \forall$ hasDaysSinceFirstVisit DatatypenegativeInteger
hasDaysSinceLastOrder	$\exists$ hasDaysSinceLastOrder DatatypeLiteral $\sqsubseteq$ Visitor
	$\top \sqsubseteq \forall$ hasDaysSinceLastOrder DatatypenegativeInteger
hasDaysSinceLastVisit	$\exists$ hasDaysSinceLastVisit DatatypeLiteral $\sqsubseteq$ Visitor
	$\top \sqsubseteq \forall$ hasDaysSinceLastVisit DatatypenegativeInteger
	$\exists$ IsReturningVisitor DatatypeLiteral $\sqsubseteq$ Visitor
	$\top \sqsubseteq \forall$ IsReturningVisitor Datatypeboolean

un número de acciones, de eventos y de búsquedas. Durante una visita se realizan transacciones. Un atributo importante de la visita es la ruta, que comprende las páginas anteriores y siguiente en el acceso a la web. De hecho, la clase "pasos de navegación" *Navigation_step* se utiliza para modelar las rutas que siguen los visitantes a lo largo de sus visitas. Los Cuadros 3 y 4 contienen las propiedades que tienen dominios las clases visitante y visita, respectivamente.

- ***Page.*** Una página en una web de comercio electrónico contiene *items*, es decir, productos y/o servicios a la venta. Los parámatros de análisis de la clase *Page* son: *average_order_value, average_time_on_page, bounce_rate, date_of_last_ visit, number_of_exits, number_of_landings, number_of_new_visitors, number_of_page_views, number_of_returning_visitors, number_of_sessions_by_medium* (los medios pueden ser: enlaces directos, redes sociales y motores de búsqueda), *number_of_sessions, number_of_unique_page_views, number_of_unique_visitors, number_of_units_sold, number_of_visitors, number_of_visits, revenue_per_session and*

Cuadro 4. Grupo Visit: las propiedades de objeto y de datos se representan en lógica de descripciones

Propiedades de Objeto	Lógica de Descripciones
hasNavigationStep	$\exists$ hasNavigationStep.Thing $\sqsubseteq$ Visit
	$\top \sqsubseteq \forall$ hasNavigationStep.Navigation_step
hasRefererKeyword	$\exists$ hasRefererKeyword.Thing $\sqsubseteq$ Visit
	$\top \sqsubseteq \forall$ hasRefererKeyword.Referer_keyword
makesVisit	hasVisitor¿ $\equiv$ makesVisit¿$^-$
	$\exists$ makesVisit.Thing $\sqsubseteq$ Visitor
	$\top \sqsubseteq \forall$ makesVisit.Visit
Propiedades de Datos	Lógica de Descripciones
hasDuration	$\exists$ hasDuration.DatatypeLiteral $\sqsubseteq$ Visit
	$\top \sqsubseteq \forall$ hasDuration.Datatypetime
hasReturningVisitor	$\exists$ hasReturningVisitor.DatatypeLiteral $\sqsubseteq$ Visit
	$\top \sqsubseteq \forall$ hasReturningVisitor.Datatypeboolean

Cuadro 5. Grupo Page: las propiedades de objeto y de datos se representan en lógica de descripciones

Propiedades de Objeto	Lógica de Descripciones
hasNumberOfVisits	$\exists$ hasNumberOfVisits Thing $\sqsubseteq$ Eshop $\sqcup$ Page $\sqcup$ Visitor
	$\top \sqsubseteq \forall$ hasNumberOfVisits Number_of_visits
hasNumberOfVisitors	$\exists$ hasNumberOfVisitors Thing $\sqsubseteq$ Eshop $\sqcup$ Page
	$\top \sqsubseteq \forall$ hasNumberOfVisitors Number_of_visitors
hasTotalRevenue	$\exists$ hasTotalRevenue Thing $\sqsubseteq$ Eshop $\sqcup$ Page
	$\top \sqsubseteq \forall$ hasTotalRevenue Total_revenue
isOnPage	$\exists$ isOnPage Thing $\sqsubseteq$ Item
	$\top \sqsubseteq \forall$ isOnPage Page
Propiedades de Datos	Lógica de Descripciones
hasName	$\exists$ hasName DatatypeLiteral $\sqsubseteq$ Browser $\sqcup$ Competitor $\sqcup$ Eshop $\sqcup$ Goal $\sqcup$ Item $\sqcup$ Operating_system $\sqcup$ Page $\sqcup$ Product
	$\top \sqsubseteq \forall$ hasName Datatypestring
hasURL	$\exists$ hasURL DatatypeLiteral $\sqsubseteq$ Competitor $\sqcup$ Eshop $\sqcup$ Page $\sqcup$ Price
	$\top \sqsubseteq \forall$ hasURL Datatypestring
hasTitle	$\exists$ hasTitle DatatypeLiteral $\sqsubseteq$ Page
	$\top \sqsubseteq \forall$ hasTitle Datatypestring

total_revenue. Los atributos de una página son básicamente el título y la URL. Cuadro 5 contiene una serie de propiedades representativas del dominio de página. Una propiedad interesante es *hasTotalRevenue*, que en esta tabla hace referencia a los ingresos generados en la propia página, aunque puede utilizarse también para toda la e-shop, ya que se puede calcular para ambos casos.

- Item. Un Item es un producto o servicio ofrecido en una tienda online. Los items específicos de una e-shop se modelan mediante una ontología de dominio específica, es decir, viajes, libros, música, etc. La tabla del Cuadro 6 describe algunas propiedades de objeto y de datos más representativas de esta clase. De acuerdo con estas propiedades, un Item tiene un precio (*hasPrice*), que es válido durante cierta fecha. Por tanto, los atributos para precio son: valor (*value*), moneda (*currency*) y fecha de validez. Los atributos para Item son su categoría y un indicador de si han sido o no eliminados de la tienda. Los productos tienen un fabricante (*manufacturer*) y entre sus atributos cuenta con: el nombre, el tipo y su disponibilidad en fecha y referencia específica. El número de artículo puede ser uno de los códigos estándares: "ASIN", "EAN" o "ISBN".

Cuadro 6. Grupo Item: las propiedades de objeto y de dato se representan en lógica de descripciones

Propiedades de Objeto	Lógica de Descripciones
hasItem	$\exists$ hasItem Thing $\sqsubseteq$ Page
	$\top \sqsubseteq \forall$ hasItem Item
hasPrice	$\exists$ hasPrice Thing $\sqsubseteq$ Item $\sqcup$ ShareProductData
	$\top \sqsubseteq \forall$ hasPrice Price
includes	$\exists$ includes Thing $\sqsubseteq$ Order
	$\top \sqsubseteq \forall$ includes Item
isOnPage	$\exists$ isOnPage Thing $\sqsubseteq$ Item
	$\top \sqsubseteq \forall$ isOnPage Page
Propiedades de Datos	Lógica de Descripciones
hasCategory	$\exists$ hasCategory DatatypeLiteral $\sqsubseteq$ Item
	$\top \sqsubseteq \forall$ hasCategory Datatypestring
hasName	$\exists$ hasName DatatypeLiteral $\sqsubseteq$ Browser $\sqcup$ Competitor $\sqcup$ Eshop $\sqcup$ Goal $\sqcup$ Item $\sqcup$ Operating_system $\sqcup$ Page $\sqcup$ Product
	$\top \sqsubseteq \forall$ hasName Datatypestring
hasItemID	$\exists$ hasItemID DatatypeLiteral $\sqsubseteq$ Item
	$\top \sqsubseteq \forall$ hasItemID DatatypenonNegativeInteger
hasQuantity	$\exists$ hasQuantity DatatypeLiteral $\sqsubseteq$ Item
	$\top \sqsubseteq \forall$ hasQuantity DatatypenonNegativeInteger

2.2. Fuentes de Datos

Como hemos mencionado anteriormente, hemos seleccionado dos tipos principales fuentes de datos provenientes de servicios de traza web o huella electrónica ofrecidos por las aplicaciones Google Analitics y Piwik. Estas herramientas fueron seleccionadas tras una serie de entrevistas y encuestas a un conjunto de más de 150 propietarios de tiendas online de diferentes países y sectores comerciales. Como resultado se generó un informe [2] donde se obtiene, entre otras conclusiones, que la mayoría de las empresas entrevistadas (68 %) utilizan Google Analytics para hacer sus análisis de visitas, seguido por Piwik (16 %) y el resto de herramientas en el mercado. No obstante, si bien Google Analitics obtiene un gran número de atributos de visita, estos dados se sirven de forma muy agregada y los valores references a comercio sólo están disponibles en su versión de pago. Por tanto, como complemento a esta fuente de información, también utilizamos la plataforma Piwik, ya que nos ofrece una gran cantidad de atributos desagregados, además de información de comercio (ventas, ganancias, etc.) de manera abierta (sin necesidad de pago).

Como fuente adicional, también se contemplan los datos que se generan mediante procesos dedicados de barrido web, conocidos como *Web Scrapping*, que obtienen información de una selección de webs de competidores para el seguimiento de precios y novedades de productos.

El proceso de recolectar los datos de diferentes fuentes y transformarlos en RDF se lleva a cabo mediante las funciones de mapeo. Para cada fuente se ofrece un conjunto diferente de métodos para obtener, armonizar y almacenar los datos en RDF de acuerdo a la ontología diseñada. De hecho, se han desarrollado un gran número de funciones de mapeo correspondientes a los atributos de análisis web, en referencia a cada clase de la ontología. Sin embargo, aquellos atributos que comparten una estructura común se han mapeado utilizando funciones genéricas, por tanto aprovechando el diseño eficiente de la ontología.

- ***Google Analytics***[4]. El proceso traza web, o web tracking, en Google Analytics se realiza mediante un "Snippet" o código de huella electrónica que proporciona una API de funciones de acceso para cada valor de atributo y métrica. Esta "huella electrónica" consiste realmente en un pequeño código de JavaScript que se aloja en el fuente HTML de la tienda online y se despliega en el servicio web donde se hospeda la e-shop. Esta pieza de código activa el tracking de Google Analytics mediante el proceso de JavaScript `ga.js/analytics.js`. Las funciones de mapeo instancian este componente para la obtención de los datos, para formatearlos inmediatamente en RDF de manera sistemática. Los atributos utilizados, así como sus combinaciones, representan un conjunto lo suficientemente representativo y amplio para cubrir los requisitos de análisis. Cabe destacar que la ontología se puede extender para considerar cualquier atributo.

- ***Piwik*** [5] es una aplicación de analítica web de código abierto que se despliega en un servicio web PHP/MySQL. El proceso de tracking web se realiza en Piwik de manera similar a Google Analytics, es decir, mediante la utilización de un trozo de código de huella electrónica alojado en la tienda virtual. En el caso de Piwik, hemos desplegado nuestro propio servicio de administración, por lo que los datos de traza web se almacenan en una base de datos relacional (MySQL). Por tanto, las funciones de mapeo consultan directamente esta base de datos y transforman los valores de los atributos en formato RDF, siguiendo la estructura lógica marcada por nuestra ontología.

- ***Web Scrappers***. Entre las aplicaciones desarrolladas en el proyecto SME E-Compass, contamos además con una serie de métodos para automatizar el "rastreo" web, mediante los cuales obtenemos información en tiempo real sobre los precios y los productos en determinadas tiendas online de competidores. Esta funcionalidad ofrece un servicio de datos sobre los competidores en formato JSON[6]. En este caso, nuestros *mappings* convierten la información desde JSON a RDF siguiendo el modelo semántico especificado por la ontología.

2.3. Repositorio RDF

Toda la información procesada se consolida en un repositorio RDF que integra las distintas fuentes de datos de analítica web. Este repositorio está conectado además con un servicio de SPARQL Endpoint, a través del cual podemos hacer consultas sobre los datos de huella electrónica y scrapping de manera deambiguada e independientemente de la fuente de origen.

Como ejemplo de acceso a los datos, consideremos un escenario en el que un algoritmo de análisis precisa de información referente a las visitas de una tienda online determinada, en cierta fecha y/o periodo de tiempo. La información requerida consistiría tanto en datos desagregados de visitas, como aquellos proporcionados por Piwik, así como en métricas calculadas, como las proporcionadas por Google Analytics.

[4] http://www.google.com/analytics/
[5] http://piwik.org/
[6] http://json.org

Consulta SPARQL Q1:

```
PREFIX
vis:<http://www.sme-ecompass.eu/ontologies/visitor_behaviour.owl#>
SELECT ?e as ?eshop, ?fat as ?date, ?vi as ?visit,
?vts as ?visit_total_searches, ?vte as ?visit_total_events,
?vd as ?visit_duration, ?vgc as ?visit_total_goal_converted,
?bv as ?total_bounce_rate, ?cv as ?total_conversion_rate,
?lv as ?total_number_of_entries, ?nv as ?total_number_of_new_visitors
FROM
<http://www.sme-ecompass.eu/ontologies/visitor_behaviour/<eshop_id>/>
WHERE{
   ?e vis:hasVisitor ?vt.
   ?vt vis:makesVisit ?vi.
   ?vi vis:hasFirstActionTime ?fat.
   ?vi vis:hasNumberOfSearches ?vts.
   ?vi vis:hasNumberOfEvents ?vte.
   ?vi vis:hasDuration ?vd.
   ?vi vis:hasTotalGoalConverted ?vgc.
   ?e vis:hasBounceRate ?b.
   ?b vis:hasValue ?bv.
   ?b vis:hasDate ?d.
   ?e vis:hasConversionRate ?c.
   ?c vis:hasValue ?cv.
   ?c vis:hasDate ?d.
   ?e vis:hasNumberOfLandings ?l.
   ?l vis:hasValue ?lv.
   ?l vis:hasDate ?d.
   ?e vis:hasNumberOfNewVisitors ?n.
   ?n vis:hasValue ?nv.
   ?n vis:hasDate ?d.
   FILTER(str(?fat) > "2015-10-23" && str(?fat)
   < "2015-10-24" && str(?d) = "2015-10-23")
}
```

La consulta SPARQL Q1 unifica la lógica de acceso a partir de la cual se obtienen los resultados de ejemplo resumidos en el Cuadro 7. Estos resultados se corresponden con dos visitas consecutivas a la tienda con ID `<eshop-id>`, que se realizaron con fecha 2015-10-23. Los IDs de las visitas son 75688 y 75692, las cuales se capturaron con marca temporal 14:19:44 y 14:21:41, respectivamente. Tal y como se muestra en la tabla, la visita con tiempo más prolongado desembocó en un objetivo de conversión (una venta efectiva), mientras que la visita de corta duración terminó sin ningún tipo de conversión, lo cual representa al visitante que abandona la tienda de manera prematura.

En el caso de los atributos agregados, éstos son calculados para todas las visitas en el periodo de tiempo establecido en la consulta SPARQL. Por lo tanto, como se muestra en la segunda mitad del Cuadro 7, la tienda registra una "tasa

Cuadro 7. Dos ejemplos del resultado de la consulta SPARQL Q1 para un determinado intervalo temporal (día 2015-10-23) de una tienda online real

Atributo/Métrica	Visit75688	Visit75692
timestamp	14:19:44	14:21:41
visit_total_searches	0	0
visit_total_events	0	0
visit_total_duration	2071	12
visit_total_goal_converted	1	0
total_bounce_rate	52.6066	
total_conversion_rate	34.1232	
total_number_of_entries	211	
total_number_of_new_visitors	145	

de rebote" (*bounce rate*, se refiere a la proporción de visitantes que entran en un sitio web y lo abandonan después de haber visto una sola página web, en unos pocos segundos) cercana al 53 %, con un ratio de conversión del 34,12 %. Estos porcentajes corresponden a todas las visitas, rebotes y compras en la tienda durante el periodo temporal especificado en la consulta. Otro atributo de especial interés es el número de nuevos visitantes, que para la tienda examinada y el periodo de tiempo consultado, es de 145, es decir, el 68.72 % sobre el conjunto total de visitas en la web. Toda esta información puede ser ahora utilizada en conjunto para entrenar algoritmos predictivos de minería de datos, con el objetivo de obtener indicios que ayuden al comerciante a adoptar una determinada estrategia de marketing para captar nuevos clientes.

En este sentido, el repositorio RDF propuesto incorpora además un servicio de funciones REST API que implementan consultas SPARQL predefinidas. De esta forma se automatiza y se simplifica el acceso a la información almacenada. Los datos proporcionados por estas funciones de consulta son entonces utilizados como entrada para los algoritmos minería de datos que realizan los procesos de análisis. En la siguiente sección, pasamos a describir un caso de uso típico de análisis del visitante en términos de validación de nuestro modelo semántico.

3. Validación: Caso de Uso

El análisis de productos permite comprender mejor la relación entre visitantes y artículos comprados, que en función de los resultados, se traducirán en actuaciones sobre el precio o el posicionamiento del producto para mejorar su visibilidad en la web. El análisis en este sentido permite conocer la relación entre los visitantes que buscan cierto producto en la tienda y qué proporción de ellos terminan comprando. Nos centramos en este caso de uso en el análisis de una tienda online real de Alemania dedicada a la venta de artículos de belleza.

La información obtenida puede ayudar a optimizar las ventas. Si un producto es raramente visitado, pero la tasa de conversión es alta, podríamos colocarlo en un lugar más prominente en el sitio web para mejorar las ventas y actuar como "gancho" para otros artículos. Un producto que tiene pocos visitantes y bajo ratio de conversión, debe ser inspeccionado en cuanto al precio o incluso sustituirlo o eliminarlo por completo.

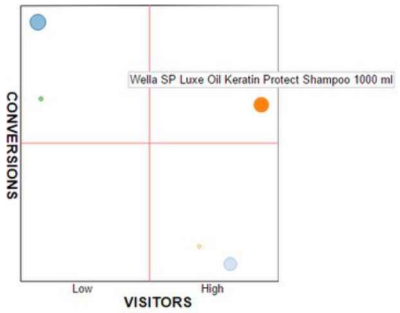

Figura 2. Análisis mediante gráfica DAFO de visitantes y productos

Este tipo de análisis requiere de información, tanto de actividad respecto a la navegación del visitante, como sobre los hábitos de compra. Por tanto, para este caso de uso nos centramos en datos capturados respecto a atributos de Piwik, como: *idaction_sku* (Stock-keeping unit), *idaction_name, idaction_category, location_geoip, visit_first_action_time, visitor_days_since_order, visit_goal_buyer* y *visit_goal_converted*. Estos atributos se modelan mediante nuestra ontología según las descripciones de la Sección 2.1 para la clase *Visit*.

El modelo de minería utiliza en primer lugar, un algoritmo de aprendizaje no-supervisado: clustering para generar grupos de visitantes y productos por comportamiento; y en segundo lugar una técnica de aprendizaje supervisado mediante árboles de decisión, para asignar los nuevos visitantes y productos a sus correspondientes grupos.

Como resultado, La Figura 2 muestra las relaciones entre conversiones de productos y visitantes. Se trata de un gráfico DAFO (Debilidades, Amenazas, Fortalezas y Oportunidades) en el cual, el eje horizontal indica la cantidad de visitantes para cierto producto, mientras que el eje vertical muestra el ratio de conversión de dicho producto. El tamaño de los puntos representan los ingresos totales producidos por el producto. Por tanto, se pueden establecer diferentes estrategias comerciales por las indicaciones de cada cuadrante. Los productos con pocos visitas y bajo ratio de conversión denotan poco interés por parte de los visitantes, con pocas ventas. Los productos más rentables se localizan en el cuadrante con pocas visitas pero alto número de conversiones, por lo que registran poco tráfico web. En este sentido, con cierta actividad promocional adicional, este producto podría obtener incluso mayores ventas. Aquellos productos con muchas visitas pero pocas ventas son candidatos típicos para revisar su posición en la web, su precio, etc.

Por último, los productos con muchas visitas y alto ratio de conversión denotan artículos atractivos y rentables (véase el ejemplo de "Wella SP Luxe Aceite Queratina Protect Champoo 1000 ml" en la figura). La estrategia global sería entonces intentar posicionar nuestros productos en el cuadrante superior de la derecha referente a productos con gran número de visitas y ventas.

4. Conclusiones

En este trabajo, proponemos un repositorio RDF para recopilar, integrar y almacenar información de traza web procedente de distintas fuentes de huella electrónica. Estas se consolidan en el repositorio diseñado para proporcionar semántica común a los datos y dar servicio homogéneo a algoritmos de Minería de Datos. El servicio propuesto se ha validado mediante traza digital real (Google Analitics y Piwik) de 15 tiendas virtuales de diferentes sectores y países europeos (UK, España, Grecia y Alemania) durante varios meses de actividad. En concreto, se presenta un caso de uso real sobre el análisis de visitas y conversiones en una tienda de productos de belleza.

Como aportación principal y original de este trabajo, hemos diseñado e implementado por primera vez una ontología OWL (Web Ontology Language) para analítica web. Esta ontología contempla un conjunto de atributos y métricas de traza web, lo suficientemente exhaustivo y complementario, provenientes de las herramientas más representativas y utilizadas hoy en día para el análisis web: Google Analytics y Piwik, además de la aplicación propia de Web Scrapping.

Como trabajo futuro, el siguiente paso consistirá en ampliar la ontología para considerar nuevas fuentes de analítica web de otros analizadores comerciales (Adobe Syte Catalyst, Yahoo WA, etc.). Además, estamos interesados en incorporar a nuestro repositorio otros conjuntos de Open Linked Data para enriquecer el modelo semántico con información que proporcione nuevas perspectivas al análisis: información meteorológica, tendencias de consumo, descripciones de productos, afinidades por sector social o geográfico, etc.

Referencias

1. M. Dean and G. Schreiber. OWL web ontology language reference. Technical report, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-owl-ref-20040210/, Latest version available at http://www.w3.org/TR/owl-ref/, 2004.
2. SME E-Compass. D2.1 sme-e-compass requirements analysis. Technical report, Public Deliverable, 2004.
3. M. Hepp. Goodrelations: An ontology for describing products and services offers on the web. In *Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW2008)*, pages 332–347. Springer LNCS, Vol 5268, 2008.
4. D. McGuinness and F. Harmelen. OWL web ontology language overview. Technical report, W3C Recommendation, 2004.
5. Natalya F. Noy and Deborah L. McGuinness. Dontology development 101: A guide to creating your first ontology. Technical report, tanford University Knowledge Systems Laboratory Technical Report KSL-01-05, 2001.