SIS: Semantic Intelligent Search Engine from heterogeneous information sources applied to e-commerce

Ana Flores Cuadrado, Eduardo Villoslada de la Torre

Telefónica Research & Development
Parque Tecnológico de Boecillo 120, 47151 Boecillo. Valladolid, Spain
{anafc, evdlt}@tid.es

Abstract: SIS is a Semantic Search Engine applied to Web-Commerce to unify and relate heterogeneous data from different web sites that removes language barriers, synonyms or currency related issues. SIS leans on a platform that combines ontologies with thesauri to improve the data retrieval and the semantic queries. The results are enhanced by unifying the vocabulary of the information retrieved and translating the queries into the ontology domain. The platform architecture is suitable for two different tasks: domain definition and use of the search engine. The present article presents SIS as a practical realization of the platform in the field of e-commerce.

1 Introduction

Internet is massively used by companies to do business. However, e-commerce is not always successful. There is a large amount of data available on the Internet and it is not always easy to find what we are looking for. The users often resort to general purpose search engines ([Fe03], [AV04]) like Google[1] [BP98], AltaVista[2], Yahoo[3] or others *Business to Consumer (B2C)* oriented like eBay[4]. But all these engines perform keyword based searches instead of semantic search engines and therefore the users are forced to know exactly the term they are searching for [AV04]. Nowadays, there is a surge of new generation semantic search engines (Hakya[5], Powerset[6]) that lean on ontologies [Gr95] to translate the natural language queries of the users and to find the relationships between keywords. But these engines are not able to unify and relate heterogeneous data, besides they are not specialized in searches for e-commerce.

In *B2C,* the search engines should also be able to compare the data retrieved from different pages. But the present shop-bots ([Fe03], [DEW98]) cannot compare them since they do not know how relate the information extracted from heterogeneous data sources. This situation aggravates due to a number of circumstances: each source has its

[1] Google: http://www.google.com
[2] AltaVista: http://www.altavista.com
[3] Yahoo: http://www.search.yahoo.com
[4] eBay: http://www.ebay.com/
[5] Hakya: http://www.hakia.com/
[6] Powerset: http://www.powerset.com/

own vocabulary, there are synonyms, or maybe there is information expressed in different languages or currencies [AV04].
On the other hand, the techniques for data extraction from Web pages ([La02],[Li07]) poses serious problems ([HKH06], [Wi06]) which have not been solved yet. To make easier the automated data retrieval from semantically annotated web sites, the *World Wide Web Consortium* (*W3C*) has produced standards that are currently used by tools such us *Piggy Bank* [HMK05] or the semantic scrappers.
Our main goal is to prove that searches and data extraction from heterogeneous web sites are more efficient combining the use of ontologies with thesauri. For this purpose we have built a platform that extracts data from pages semantically annotated from different web sites. The platform unifies the data format and stores the information onto an ontological store that allows semantic searching. The platform architecture is based on the Ontobroker project [Fe03]. We have incorporated new features to the platform such as combined use of ontologies and thesauri or the use of techniques for semantic data annotation and extraction. Additionally, we have extended the architecture with a multilevel design to isolate the platform operation from the domain definition.

In Section 2 there is a general description of the main components of the platform. In Section 3 we introduce the application of the platform to e-commerce. Finally, we conclude with the pros and cons of the platform and ideas for future work.

2 Platform Description

2.1 Extraction Subsystem

The subsystem comprises 3 modules (Figure 1). The **Knowledge Acquisition** module looks for the information on the web pages, to extract it and transform it by adding semantic information. This module stores the data in a *Resource Description Framework (RDF*[7] [AV04]*)* file, compatible with the ontology. This module only processes pages with semantic marks, provided that the pages are in accordance with the protocol *Gleaning Resource Descriptions from Dialects of Languages (GRDDL*[8]*)*. The algorithm to extract the semantic data (usually an *Extensible Stylesheet Language Transformations (XSLT)*[9]*)* file*)* is established by the *GRDDL* protocol. The semantic marks can be represented with *microformats*[10] *or RDF Attribute (RDFa)*[11] . Thus, algorithm and *microformat* are independent of the domain of the data under process. The **Ontology Population** module homogenizes and classifies the contents of the *RDF* store. Afterwards, the results of this process finally populate the ontology *(Web Ontology Language (OWL)* [12][AV04]). The homogenization process unifies the format of the

[7] RDF: http://www.w3.org/TR/rdf-primer/; http://www.w3.org/TR/rdf-syntax-grammar
[8] GRDDL: http://www.w3.org/TR/grddl/
[9] XSLT: http://www.w3.org/TR/xslt
[10] Microformats: http://microformats.org/
[11] RDFa: http://www.w3.org/TR/xhtml-rdfa-primer/; http://www.w3.org/TR/rdfa-syntax
[12] OWL : http://www.w3.org/TR/owl-ref/

information extracted within a single vocabulary by means of conversions (currency conversion), transformations (correction of typing mistakes; use of synonyms) and translations using thesauri - *Simple Knowledge Organization System (SKOS[13]).* In the course of the classification, specific domain relations and negation constrains are added. In addition, the module applies an update policy to make the new information consistent with the previous data. (In the event of a conflict, the policy sets up whether the new information will be considered as new data instances or if the existing ones will be updated). **The inference module** uses reasoners (with *Description Logic Interface (DIG*[14]*))* to infer additional knowledge over the populated ontology. The result is stored on a semantic store *(OWL)* accessible by *Hypertext Transfer Protocol (HTTP).*

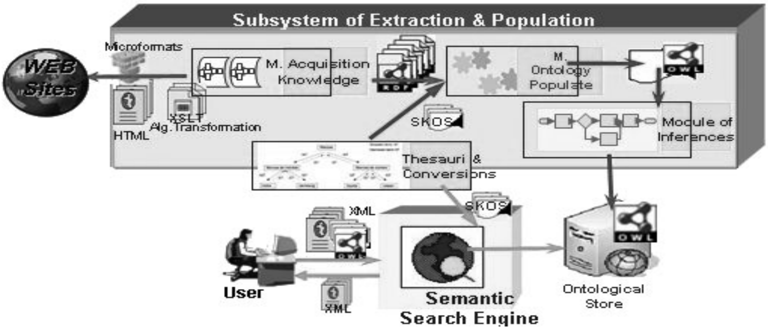


Figure 1: Platform Architecture

2.2 Web Semantic Search Engine

The Semantic Search Engine empowers the users to search on an ontological store accessible via *HTTP.* The process combines semantic searches (where users are guided by the ontology), with use of conversions and thesauri to transform the queries from the users' vocabulary to that of the data store. The Web Semantic Search Engine is generical and operates independently from the ontology domain and the store queried. The user selects the ontological store to query on the **Web Interface** first and then a screen dedicated to searches appears. This screen helps the user to construct the query, choosing the concept to look for, and adding constrains to the values of the attributes. The fields of this screen are dynamically displayed based on the ontology of the store and the selections performed by the user. The users may also add more complex constraints on this screen by navigation through relations between two, three or more ontology concepts, and adding constraints on attributes values of the related instances. The query is sent to the **Search Engine** with an *Extensible Mark-up Language (XML)*[15] [AV04] message**.** The **Search Engine** may modify the query by applying thesauri or conversions on the values of the attributes of the constraints. Besides, the **Engine** adds other

[13] SKOS: http://www.w3.org/TR/skos-reference
[14] DIG: http://dl.kr.org/dig/
[15] XML: http://www.w3.org/XML//

conditions to the query to obtain subclass instances of the class searched and also to search the instances with unknown values in any of its attributes. Then the query is translated into *SPARQL Query Language for RDF (SPARQL)* [16] and the results are sent to the **Web Interface** in an *XML* message conforming to the standard [17].

3 A Practical Case for Web-commerce

This case is based on semantic searches of mobile phones. The attributes of the mobile phones are extracted from different sites of specialized shops. In each shop, the language and the vocabulary are different, and the prices are supplied in different currencies. Besides, each online shop is somehow associated to a number of points of sale physically located in different addresses.

3.1 Ontology and Vocabularies

Many authors have made in the past an attempt to define a unified ontology for e-Business based on the main e-commerce standards ([Do01],[Fe03], [GFC03]). Also, new initiatives have recently appeared (*Electronic Business XML (ebXMLl)*[18] *Registry for OWL, Universal Business Language (UBL) Ontology*[19]*, Universal Standard Products and Services Classification (UNSPSC)* [20]*, International Standard for the Classification and Description of Products and Services (eCl@ss*[21]*)).* Some of these initiatives such as the UNSPSC or the eCl@ssOwl do not have important attributes for e-business (like the price or VAT), and the others such as the UBLOntolgy has already been developed. For these reasons, we deem it advisable to create an ontology for testing the performance of the platform. We have used *Methontology* [GFC03] to build the axioms of the ontology in our *B2C* domain: *(Mobile Phone ⊒ Fixed Phone ⊒ Product); (Mobile ≡ (≥ 1 is_the_model)); (Model ≡ (= 1 has_tradeMark)); (Product ≡ ((≥ 1 has_price) ∩ (≥ sold_in_shop))); (Price (≥ 1 is_sold_in)); (Shop ≡ (≥ 1 has_retail_price)); (SalePoint ≡ (=1 located_on)*.

3.2 Example of extraction and query

For our example we use a mobile phone with *Sony Erics* as trademark and *V630iBlanco* as model. with *Bluetooth*, *Mp3* and *3G* and it is sold in *Vodafone* shops for *59 euros*.

During the process of extraction, the **Acquisition module** processes only shop pages where **microformat** *product odd* appears and that conform to the protocol *GRDDL* with

[16] SPARQL: http://www.w3.org/TR/rdf-sparql-query/

[17] SPARQL -XML: http://www.w3.org/TR/rdf-sparql-XMLres/

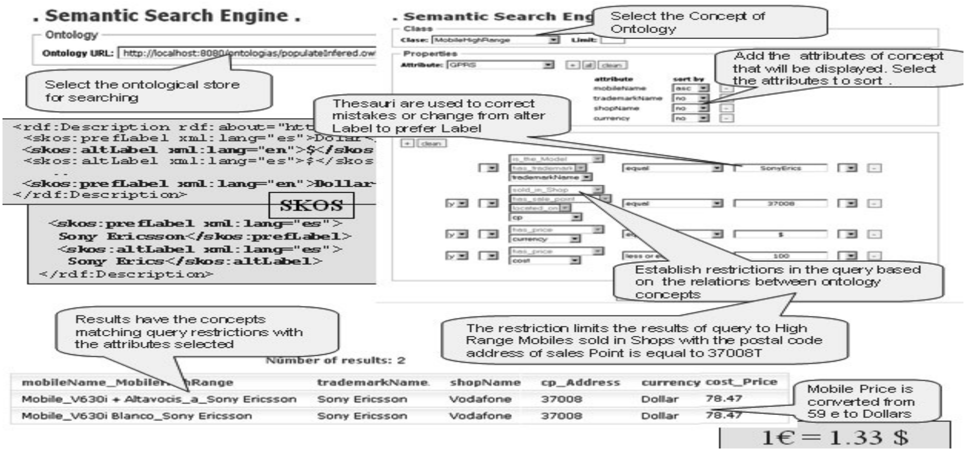
[18] ebXML Registry Profile for Web Ontology Language OWL: http://www.oasis-open.org/committees/download.php/17042/regrep-owl-profile-1.0-draft%204.pdf

[19] UBLOntolgy: http://ontolog.cim3.net/cgi-bin/wiki.pl?UblOntology

[20] *UNSPC*: http://www.eccma.org/unspsc/browse/ ; http://www.cs.vu.nl/~mcaklein/unspsc/

[21] eCl@ssOwl: http://www.heppnetz.de/eclassowl/

the location of *XSLT* algorithm that will be used for the extraction. This algorithm contains rules to indicate that mobile phone attributes are labeled in *HTML* and that should be used by **Acquisition module** to generate the *RDF* file. The **Population module** creates mobile phone instances with the data of the *RDF* file. Prior to storing them on the ontological store, the **Population module** looks for the alternative terms of our thesauri *(SKOS)* in the attribute values and changes those values by their preferred terms in the thesauri. Following our example, the **Population module** replaces the trademark value *Sony Erics* by *Sony Ericsson*. The **Population module,** also, applies conversions to change the price from *59 euros* to *78.47 dollars* and. it adds several additional constraints such as “The mobile is sold in a shop”, or “The shop has several point of sale”. Furthermore, we have defined some rules to enable the inference engine to classify the mobiles in the *MobileHighRange* class if they come with *3G or Bluetooth*, *Mp3*, *Camera* and *Radio.*



Ontology URL: http://localhost:8080/ontologias/populateInfered.ow

Select the ontological store for searching

```
<rdf:Description rdf:about="htt
  <skos:prefLabel xml:lang="es">Dolar<
 <skos:altLabel xml:lang="en">$</skos
  <skos:altLabel xml:lang="es">$</skos
    ..
 <skos:prefLabel xml:lang="en">Dollar
</rdf:Description>
```

```
  <skos:prefLabel xml:lang="es">
   Sony Ericsson</skos:prefLabel>
  <skos:altLabel xml:lang="es">
   Sony Erics</skos:altLabel>
 </rdf:Description>
```

Select the Concept of Ontology

Clase: MobileHighRange

Attribute: GPRS

attribute	sort by
mobileName	asc
trademarkName	no
shopName	no
currency	no

Add the attributes of concept that will be displayed. Select the attributes t o sort .

Thesauri are used to correct mistakes or change from alter Label to prefer Label

is_the_Model / has_trademark / trademarkName — equal — SonyErics

sold_in_Shop / has_sale_point / located_on / cp — equal — 37008

has_price / currency — equal — $

has_price / cost — less or equal — 100

Establish restrictions in the query based on the relations between ontology concepts

The restriction limits the results of query to High Range Mobiles sold in Shops with the postal code address of sales Point is equal to 37008T

Results have the concepts matching query restrictions with the attributes selected

Number of results: 2

mobileName_MobileHighRange	trademarkName	shopName	cp_Address	currency	cost_Price
Mobile_V630i + Altavocis_a_Sony Ericsson	Sony Ericsson	Vodafone	37008	Dollar	78.47
Mobile_V630i Blanco_Sony Ericsson	Sony Ericsson	Vodafone	37008	Dollar	78.47

Mobile Price is converted from 59 e to Dollars

1€ = 1.33 $

Figure 2: Constraints and results for Semantic Search

In Figure 2 we want to search *MobileHighRange* with the manufacturer *Sony Ericsson* and a price lower than *100 $.* Th*ey* must be sold in a shop with the 37008 as postcode. In order to narrow down the address of the point of sales we have navigated through the relationship *sold_in _shop* (between *Product* and *Shop* classes), by the relationship *has_sale_point* (between *Shop* and *SalePoint)* and by the relationship *located_in* (between *SalePoint* and *Address)*. Finally, in the class *Address* the value of the *pc* attribute is constrained to *37008*. The manufacturer an price constraint are similar cases of relationship navigation. In the manufacturer case, the user makes a mistake in the manufacturer, and type in “*Sony Erics”*. The **Search Engine** uses the thesauri to correct it to *Sony Ericsson*, before searching. In the case of the price constraint, the **Search Engine** uses the thesauri to translate the text of the currency from *$* to *dollar*. The search results (Figure 2) contain the mobile phone “*Sony Ericsson 630Vi Blanco”* but with the price in dollars.

4 Conclusions and Future Work

In this article we present a semantic platform for searching data from heterogeneous information sources and its application to e-commerce. The platform uses the combination of ontologies and thesauri to transform the heterogeneous information to a vocabulary compatible with the ontology. The transformation process puts in relation information from different sources or unifies synonymous terms, and translates them into an only unique language. Additionally, it applies conversions and corrections of typing mistakes. The information transformed is stored on an ontological store, after applying inference. Users may make semantic searches on an ontological store using Web browsers and they are guided in the search by concepts and the ontology structure. It allows more complex queries, using navigation through the relationships between classes to constraints. The users' queries are translated using thesauri to improve the search results. The platform architecture has two levels, decoupling the system operation and the domain definition. This is why the platform may be used in other fields.

In the future, the platform might give support to input data marked with *RDFa.*. Also, the information might come via Web Services calls. Hence, the transformation process would be independent of the extraction mechanism. It is possible to use Web Services to search in the semantic store. The user and the search engine might exchange the query and the results wrapped on *Simple Object Protocol Access* (*SOAP)* messages. Other possibility could be to consider user preferences as a constraint for the searches. And eventually we might use domain rules shared between systems for the inference process.

Bibliography

[AV04] Antoniou, G; van Harmelen, F: The Semantic Web Primer. MIT Press, 2004.

[BP98] Brin, S. Page L: The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems. 1998.

[DEW98] Doorenbos, R., Etzioni, O, and Weld, D.S.: A Scalable Comparison-Shopping Agent for the World-Wide Web . In Autonomous Agents Conference Proceedings, 1997.

[Do01] Dörr, M., et.al: State of the Art in Content Standards, OntoWeb Deliverable 3.1. 2001

[Fe03] Fensel, D: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce: 2nd Edition, Springer. 2003

[GFC03] Gómez-Pérez, A; Fernández-López, M; Corcho, O: Ontological Engineering. Springer-Verlag, London, 2004.

[Gr95] Gruber, T. R. Toward principles for the design of ontologies used for knowledge sharing. *International Journal Human-Computer Studies* 1995,

[HKH06] Holzinger, W; Krupl, B ; Herzog , M: Using Ontologies for Extracting Product Features from Web Pages. Semantic Web Conference (ISWC). Springer. 2006

[HMK05] Huynh, D; Mazzocchi, S; Karger, D: Piggy Bank: Experience tche Semantic Web Inside Your Web Browser. Semantic Web Conference (ISWC). Springer-Verlag 2005

[La02] Laender, A., et.al: A brief survey of web data extraction tools. SIGMOD. 2002.

[Li07] Liu, B. Web Data Mining: Exploring Hyperlinks, Contents and Usage Data. Springer- Verlag, Berlin 2007

[Wi06] Wilson, T. Three common methods for data extraction. 2006. http://blog.screen-scraper.com/2006/03/21/three-common-methods-for-data-extraction/