MORIARTY: IMPROVING ‘TIME TO MARKET’ IN *BIG DATA* AND *ARTIFICIAL INTELLIGENCE* APPLICATIONS

P. PEÑA[1], R. DEL HOYO[1], J. VEA-MURGUÍA[1], V. RODRIGÁLVAREZ[1], J.I. CALVO[1]&J.M. MARTÍN[2]
[1]Software Engineer group of Technological Institute of Aragón, Spain.
[2]Research and Development Department of INYCOM, Spain.

ABSTRACT

The objective of this paper is to present the Moriarty framework and show one use case of the recommendation of entertainment events. Moriarty is a tool that can generate *Big Data* near real-time analytics solutions (*Streaming Analytics*). This new tool makes possible the collaboration among the data scientist and the software engineer. Through Moriarty, they join forces for the rapid generation of new software solutions. The data scientist works with algorithms and data transformations using a visual interface, while the software engineer works with the idea of services to be invoked. The underlying idea is that a user can build projects of Artificial Intelligence and Data Analytics without having to make any line of code. The main power of the tool is to reduce the ‘time to market’ in an application which embeds complex algorithms of Artificial Intelligence. It is based on different Artificial Intelligence algorithms (like *Deep Learning*, *Natural Language Processing* and *Semantic Web*) and *Big Data*modules (Spark as a distributed data engine and access to *NoSQL* databases). Moriarty is divided into several layers; its core is a *BPMN* engine, which executes the processing and defines data analytics process, called *workflows*. Each workflow is defined by the standard *BPMN model* and is linked to a set of reusable functions or Artificial Intelligence algorithms written following a service-oriented architecture. An example of service presented is a recommendation application of restaurants, concerts, entertainment and events in general, where information is collected from social networks and websites, is processed by *Natural Language Processing*algorithms and finally introduced into a graph database.

Keywords: artificial intelligence, big data, moriarty, semantic,spark, streaming analytics, user profiling, workflows.

1 INTRODUCTION

The proliferation of web pages, audio and video streams, tweets, blogs and data warehousesis generating a massive amount of complex and pervasive digital data. Efficient means are now available for creating, storing and sharing this information, which also fuels data growth. However, extracting useful knowledge from huge digital datasets requires smart and scalable analytics services, programming tools and applications.

This work introduces a *Moriarty* framework that allows designing and implementing advanced Artificial Intelligence software solutions. It provides the means to address key areas of concern for business using Big Data. With Moriarty is possible to understand and structure information, identify hidden patterns and correlations in the data, and induce knowledge, as well as build learning systems. It is a flexible, precise and simple way to turn data into valuable information for strategic decision-making. The tool separates the scientific data that works with algorithms and data transformations using a visual interface, from the software engineer that works with the idea of services to be invoked. However, with Moriarty, they will collaborate for the rapid generation of new Big Data near real-time analytics solutions. Moriarty, based on different Artificial Intelligence techniques (*Deep Learning*, *Natural Language Processing*, *Semantic Web*) and *Big Data* components (Spark or*NoSQL* databases)

This paper is part of the Proceedings of the International Conference on Big Data (Big Data 2016)
www.witconferences.com

ISSN: 1755-7437 (paper format), ISSN: 1755-7445 (online), http://www.witpress.com/journals
DOI: 10.2495/DNE-V11-N3-230-238

has the significant feature to reduce the 'time to market' in applications which embed complex algorithms of Artificial Intelligence. It has the capacity to be flexibly adapted to develop new functionalities through the definition of workflows, which address the complexity of scientific and business applications. Moreover, semantic analyses techniques and the rapid development of cloud services give it a differentiating value. With Moriarty, it is possible to obtain highly useful information through the capture, storage, process and analyze of a massive and variety amount of data.

This paper is organized as follows: Real-time analytics solutions in Big Data are presented in Section 2. Section 3 describes the advanced Artificial Intelligence software Moriarty framework for Big Data. Its architecture is explained in Section 4. An example of application based on leisure and entertainment recommendations built using Moriarty is presented in Section 5. Finally, Section 6 describes conclusions of our work and discusses ideas for future work.

2 REAL-TIME ANALYTICS SOLUTIONS IN BIG DATA

Most definitions of Big Data focus on the size of data in storage. Size matters, but there are other important attributes of Big Data, namely data variety, data velocity and value. The fourth Vs of Big Data (volume, variety, velocity and value) constitute a comprehensive definition, and they bust the myth that Big Data is only about data volume. In addition, each of the fourth Vs has its own ramifications for analytics. Again, Big Data analytics is where advanced analytic techniques operate on Big Data. The definition is easy to understand. According to a recent Gartner research, 73% of surveyed organizations either have invested in Big Data already (40%) or have plans to invest within 24 months (33%) [1]. They are investing or planning to invest in Big Data innovative solutions, especially in pilots and experiments, leaded by innovation teams and research institutes. The most important challenges are the Return of Investment (ROI), the necessity of real production uses cases in companies, the skills required to deploy these uses cases and the need of a European regulatory framework for data analytics.

Big Data analytics can be explained as the application of data processing techniques to discover patterns, extract knowledge and gain insights from large-scale, typically multi-source data collections that may contain structured, unstructured and semi-structured data. The main techniques in this area are statistical, machine learning, text mining and natural language processing.

Currently in the market, there are different tools (normally from the Data mining and Business Intelligent world) that provide advantages in different aspects of Big Data Analytics and Artificial Intelligence (AI) but not in a holistic way. From Business Intelligent perspective you can find Tableau, Pentaho and qlinkview. They provide the ability to query relational and cloud databases, cubes and spreadsheets and then generate a number of graph types that can be combined into dashboards and shared over a computer network or the Internet. On the other hand, Apache Spark is a fast and general engine for large-scale data processing. MLlib is Spark's Machine Learning Library. Its goal is to make practical machine learning scalable and easy. A different alternative is H_2O, focused on Machine Learning algorithms but lacks a Web GUI and methods to simplify the analytics design. The Caffe, Torch, Theano or Tensorflow frameworks are designed to let researchers create and explore Deep Neural Networks (DNNs). However, these tools are only focused on one specific AI technique DNN. Finally, commercial solutions like SAP or Rapidminer are closer to the idea of Moriarty but are focused only on the data mining or in some cases on text mining, but lack in graphs, rule engines or knowledge representation algorithms based on ontologies.

The strongest innovation of Moriarty is the possibility to mix AI techniques in data and text mining with other techniques like ontologies, graph theory and knowledge representation, in a simple web interface, making simple the development of complex Big Data analytics software services.

In addition, the focus of Moriarty is to make possible collaboration between data scientist and software developer. None of these tools previously named are defined to develop real business solutions, only as tools of data analysis. Certainly, any of them are defined as collaboration tool to speed up the development of AI software solutions in Big Data.

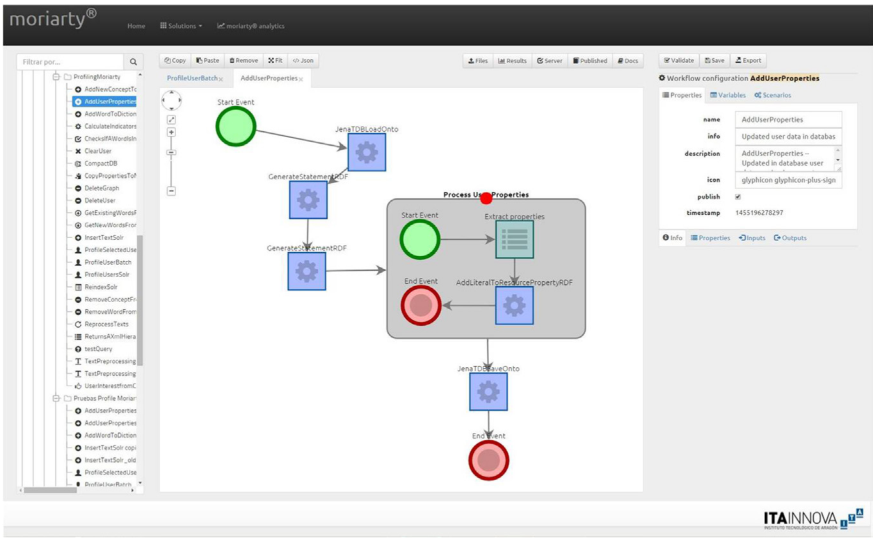
This is a clear business orientation that gathers all the AI-related knowledge in one single tool which may be exploited by a data scientist (a consultant) without writing a line of code.

3 MORIARTY

Moriarty is an advanced Artificial Intelligence software solution framework for Big Data, developed by ITAINNOVA [2]. It allows understanding and structuring information, identify hidden patterns and correlations in the data, and induce knowledge, as well as build learning systems. It is a flexible, precise, and simple way to turn data into valuable information for strategic decision-making. Its capacity to be flexibly adapted to develop new functionalities through the definition of workflows, new Semantic Models and the rapid development of cloud services gives it a differentiating value. With Moriarty, it is possible to obtain highly useful information through the capture, storage, process and analyze of a massive and variety amount of data. Moriarty is the result of more than 10 years of work in Artificial Intelligence in ITAINNOVA.

This new tool, which can design and generate Big Data near real-time analytics solutions (Streaming Analytics), separates the scientific data from the software engineer. Through Moriarty, they will collaborate for the rapid generation of new solutions. The data scientist works with algorithms and data transformations using a visual interface (Fig. 1), while the software engineer working with the idea of services to be invoked. Without having to write any line of code, Artificial Intelligence and Data (and text) Analytics solutions can be built. Moriarty has the relevant and strong feature to reduce the 'time to market' in an application which embeds complex algorithms of Artificial Intelligence.

Moriarty is implemented in Java due to the existence of a large number of Artificial Intelligence algorithms implemented in this language. Moreover, this language is used extensively at enterprise level. Moriarty maximizes the use of libraries implemented, generates rapidly decoupled applications, allows adding new algorithms easily and making changes in the processes generated in a simple way, which means reduce costs. In addition, it allows deploying services as REST Services.



Workflow configuration **AddUserProperties**

name: AddUserProperties
info: Updated user data in databas
description: AddUserProperties -- Updated in database user
icon: glyphicon glyphicon-plus-sign
publish: ☑
timestamp: 1455196278297

Figure 1: Visual interface.

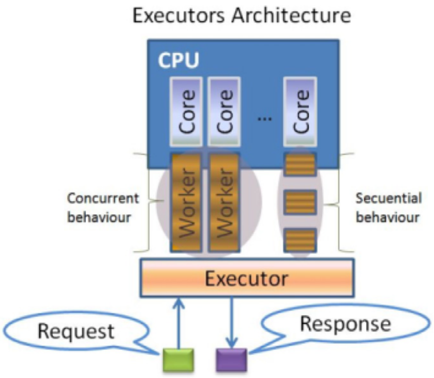


Figure 2: Executors model.

4 ARCHITECTURE

The Moriarty framework is divided into several layers. Its core is a BPMN orchestration engine, which executes the transformation or performs data analytics, called *workflows*. The engine is responsible for orchestrating the processing flows in Spark or within a multithreaded J2EE server.

Moriarty architecture consists of aworkflows engine, the *Worker* where workflows are described and the *Executor* that manages executions (Fig. 2). In particular, rules for the functions to be executed are included, organized and established in a workflow. The *Worker* focus on describing how the execution of each of the running instances of a process (workflow) is performed. The load of required workflows and libraries, the set of parameters and statistical information retrieval of execution are some common tasks of the Worker component. The Executor manages the stack of pending executions. When there are free resources for the execution of new processes, delegates the load of a workflow in the workflows runtime in the Worker. In addition, the Executor reads the interface defined by the Worker and exposes this interface as a REST service to the invocations of the workflow.

Moriarty, at runtime, initiates the Executor, which is responsible for reading the configuration files of each Worker. These contain the definition of the interface of the workflows. The Executor generates the REST interface and awaits the arrival of a request. When a process is invoked, it is checked that there are resources available on the system. An instance of a Worker, which is responsible for invoking the workflows engine with the specified process and its parameters, is created. The Worker is responsible to be waiting for the completion of the process and display execution statistics. When the process finalized, it notifies the Executor the state of the task and data returned by the execution of the workflow.

Moriarty is designed to generate service-oriented Artificial Intelligence solutions (Fig. 3). It allows users very quickly to use and compose services generated by the tool. That is, unlike object-oriented architectures, Moriarty is composed of decoupled and interoperable services. In order to create libraries and services, each workflow is defined by the BPMN standard [3] and is linked to a set of reusable Artificial Intelligence features or algorithms called workitems. jBPM is used as runtime engine of the workflows defined [4].

Moriarty contains a catalog of workitems. The workitems are libraries that can be used to build workflows through a graphical interface.Currently, more than 150 workitems developed are available, which includes both Machine1 Learning algorithms and Natural Language Processing algorithms or Web Semantic algorithms. The tool provides workitems related to the access a databases (MySQL, MongoDB, Cassandra, Virtuoso, JenaTDB), collective knowledge databases (Wikipedia, OpenDNS,

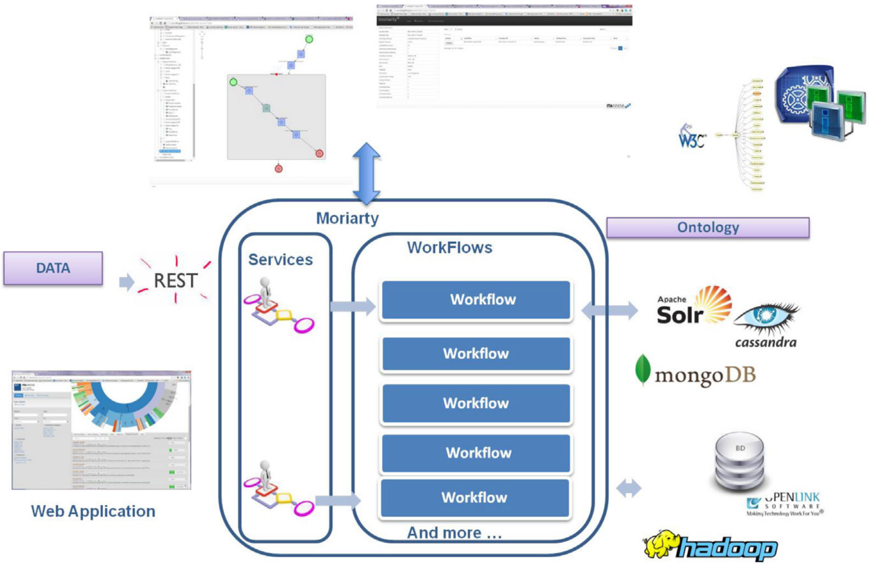


Figure 3: Stand alone server architecture.

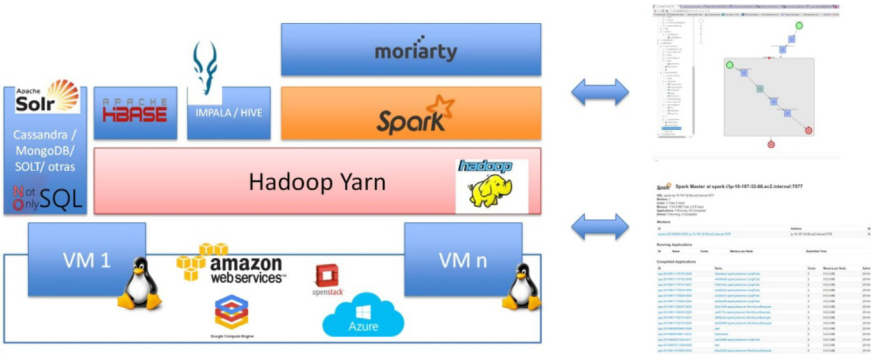


Figure 4: Distributed architecture big data.

IPTC) and social networks (Twitter, Facebook, Foursquare), to process data by natural language algorithms (steeming, lemmatized, stop words…), to use rules engines (JRules) and Artificial Intelligence reasoning (Pellet, Hermit), to management RDF triples and ontologies, etc.

Moreover, Moriarty workflows are not only available for use by external invocation, but also can be reused internally to create complex algorithms and near real-time analytics solutions. They are equivalent to the pipeline of Spark, generalizing the concept through the semantic richness offered by BPMN, to reuse it in a different environment of Spark if it was necessary. The workflows are service-oriented; this means that each service is running a business task or use case that a customer requires. When workflows are published automatically generate a REST interface that can be consumed by the final application interface.

When Moriarty is deployed in a Big Data cloud, the workflows are executed against the Spark infrastructure trough the same Web interface (Fig. 4):

5 EXAMPLE-BASED LEISURE AND ENTERTAINMENT RECOMMENDATIONS

With the increase in the amount of information about events and activities, it becomes almost impossible to discover all events of interest and select the most appropriate ones, particularly for a person with limited time. In this context, following section presents an application built using Moriarty that aims to build a semantic-based intelligent recommender system to improve end users' quality of leisure and entertainment experience. It can not only provide recommendations for group activity based on the profiles and the limitations of each member of the group but also it helps end users to organize a group activity. An optimized planner can schedule a series of recommended activities for end users and groups to get the most out of their reduced leisure time.

As it is shown in Fig. 5, this application, called Magician, gathers leisure and entertainment domain specific real and no-real time information from diverse data sources such as Twitter, Facebook, Ticketea and Foursquare in order to make intelligent recommendations based on user/group profiles.

A user profiling defined as the deduction (inference) of the interests, intentions, characteristics, behaviors and preferences, is nowadays one of the most important keys in personalized services. In recent years, important research efforts have been carried out with a wide variety of approaches, techniques and methods on the problem of how to model the interests and intentions of users to provide personalized information (filtering and sorting systems, recommendation services, navigation and search). However, it is difficult to cover all individual interests and intentions and capture the changes of user profiles. Building effective user profiles requires they evolve (human behavior is often erratic). In this context, input data can contribute to generate automatic user profiles in order to explore their needs and requirements. For this reason, Magician introduces significant improvements over today's

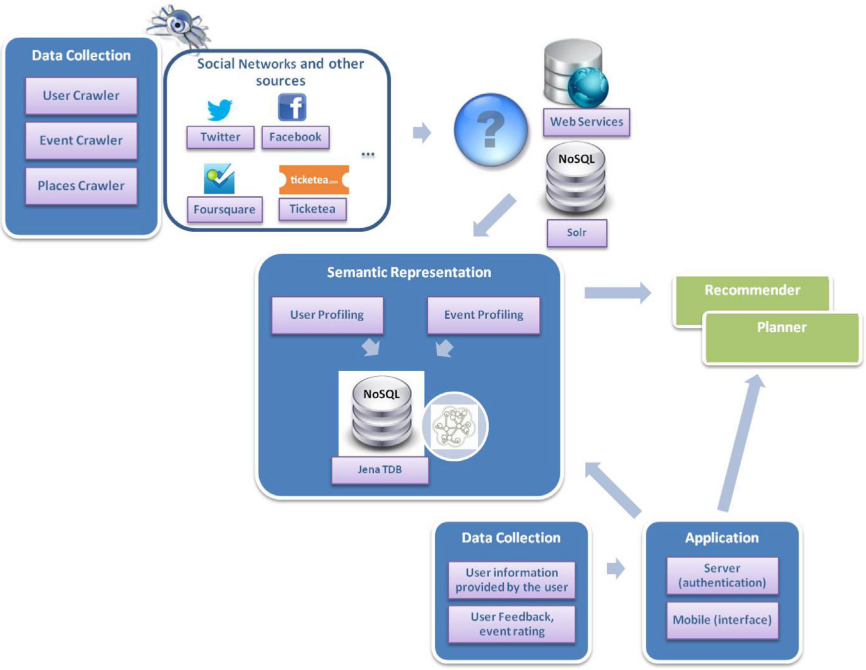


Figure 5: Magician high-level architecture and its components.

leisure and entertainment proposal solutions which act as a 'search engine specific to events and activities' and generate static and non-personalized recommendation without real-time observations of events. The introduction of ontologies allows bridging the gap between the user/group interests or profiles and the information relevant to leisure and entertainment activities (events, places, restaurants, venues, concerts, movies, sports games, weather conditions…). A user profiling [5] has been developed using Moriarty.

The definition of an ontology describing the domain of application is a key point for creating knowledge extraction algorithms and semantic annotation of unstructured data. According to one of the most widely accepted definitions of researcher Robert Thomas Gruber [6], ontology is 'an explicit and formal specification of a shared conceptualization'. To represent the collected information and user profiles in Magician, semantic technologies [7, 8]have been widely used to define and create a generic domain specific ontology using a process that follows the methodological guidelines in [9]. This ontology generated is based on existing standards taxonomies and ontologies [10–23].

The main concepts described in our ontology are related to *Person* (identification, age, gender, education,…), *Magician events* (items and activities that can be recommended to users; for example a live rock concert, taking place this night, a restaurant,…), *Interest* (items or activities in which a user is interested like rock, music, museum), *Intent* (items or activities in which a user has intention to do or visit),*URLs* (websites posted by users and extracted from social networks), *Unknow* and *Unknow category* (URLs and categories do not exist in collective knowledge repositories such as OpenDNS, DBPedia or IPTC). User preferences have been mapped to the basics properties or relationships 'hasInterest' and 'hasIntent' also defined in ontology. Inference is done to obtain the interests and intentions of each user from URLs posted in social networks. This ontology may evolve continuously depending on the new concepts that users determine by their interests and intentions and on updating collective knowledge repositories used.

Through Moriarty the logic and key algorithms based on Natural Language Processing (NLP) are implemented to populate the ontology of Magician application that models the users, the items/activities to recommend to the users, the context of the users and the relations among them with information collected from social networks and website in general.

The need to store and process huge ever-growing amounts of unstructured data gathered on the World Wide Web and to managehundreds or even thousands of operations per second from users who can demand a recommendation, lead to the adoption of a new varieties of non-relational databases, commonly references as NoSQL. The loss of flexibility or rigid schemes, the inability to scale data, the high latency or low performance and cost of RDBMS, are some of the major data management problems leading to the adoption of these technologies, nowadays widely used by companies such Amazon or Google. Two NoSQL repositories have been selected for Magician: Jena TDB[24], a graph database to store a user profile ontology model that can be queried through SPARQL [25], and Solr[26] that stores public information crawled in web sites and social networks (events, weather,…).

Magician application, running on mobile and/or tablet devices using Android OS, is an example of Big Data and Artificial Intelligence solution generated rapidly with Moriarty that allows users to create profiles (sign up), provide explicit/implicit feedback on items and more general concepts (e.g. *Brad likes rock music*), view/modify their current profile (list their friends, activities in the past,…), share information with friends, create items (e.g. *Sam organizes a soccer game in the afternoon*), get recommendation on which items to visit and view information about items (rating, photos, opening hours, address,…).

6 CONCLUSIONS AND FUTURE WORK

Moriarty is an advanced software tool that allows the data scientist and software engineer collaborate to design and generate rapidly Big Data near real time analytics solutions (Streaming Analytics). With no lines of code to write, Artificial Intelligence and Data applications can be built. Using a visual interface, data scientist works with algorithms and data transformations, while the software engineer works with the idea of services to be invoked.

Moriarty provides workitems to access a databases, collective knowledge bases and social networks, to process data by natural language algorithms, to management RDF triples, to use Artificial Intelligence reasoning, etc. The use of these workitems or libraries in a graphical way to create near real-time analytics solutions through the definition of workflows with embeds complex algorithms of Intelligence Artificial has shown that it is possible to improve significantly the 'time to market' in Big Data and Artificial Intelligence applications like Magician.

Our ambition for the future work is to further investigate in *Deep Learning* algorithms presented as one promising avenue of research into the automated extraction of complex data representations (features) at high levels of abstraction. The analysis and learning of massive amounts of unsupervised data make it a valuable tool for Big Data Analytics where raw data is largely unlabeled and uncategorized.

ACKNOWLEDGEMENTS

The dissemination of this work has been partly financed by the FSE Operative Programme for Aragon (2014–2020).

REFERENCES

[1] Survey Analysis: Big Data Investment Grows but Deployments Remain Scarce in 2014, available at https://www.gartner.com/doc/2841519/survey-analysis-big-data-investment

[2] ITAINNOVA, Instituto Tecnológico de Aragón,available at http://www.itainnova.es/

[3] Business Process Model and Notation, available at http://www.bpmn.org/

[4] Business Process Management – Process engine, available at http://www.jbpm.org/

[5] Collective Knowledge Ontology User Profiling for Twitter – Automatic User Profiling. Paula Peña, Rafael Del Hoyo, Jorge Vea-Murguía, Carlos González, Sergio Mayo. In *Proceedings ofthe2013IEEE/WIC/ACMInternational Conferenceon WebIntelligence (WI) and Intelligent Agent Technology (IAT)*.

[6] Gruber, T.R., Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human and Computer Studies*, **43**, pp. 907–928, 1995. http://dx.doi.org/10.1006/ijhc.1995.1081

[7] Web Ontology Language, availabel at http://www.w3.org/2001/sw/wiki/OWL

[8] Resource Description Framework (RDF), available at http://www.w3.org/RDF/

[9] Noy, N.F. & McGuinness, D.l., *Desarrollo de Ontologías-101: Guía Para Crear Tu Primera Ontología*. StandfordUniversity: Estados Unidos, 2005.

[10] OpenDNS cloud websites tagging,available at http://community.opendns.com/domaintagging/

[11] The Friend of a Friend ontology, available at http://xmlns.com/foaf/spec/index.rdf

[12] OWL-Time ontology, available at http://www.isi.edu/~hobbs/owl-time.html

[13] The Climate and Forecast features, available at http://www.w3.org/2005/Incubator/ssn/ssnx/cf/cf-feature

[14] PROTON (PROTo ONtology) Home Page, available at http://proton.semanticweb.org/

[15] SKOS Simple Knowledge Organization System Reference, available at http://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html

[16] wgs84_pos ontology, available at http://www.w3.org/2003/01/geo/wgs84_pos

[17] DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia, available at http://dbpedia.org

[18] Intel LEO Activities Ontology, available at http://intelleo.eu/ontologies/activities/spec/

[19] The Web Ontology for Products and Services, available at http://www.heppnetz.de/projects/eclassowl/

[20] RECO: a vocabulary to formalize preferences in the Semantic Web, available at http://ontologies.ezweb.morfeo-project.org/reco/spec

[21] International Presss Telecommunications Council, available at http://www.iptc.org/site/Home/

[22] DAML event ontology, available at http://daml.umbc.edu/ontologies/ittalks/event

[23] Event Ontology, available http://motools.sourceforge.net/event/event.html

[24] Apache Jena – TDB, available at http://jena.apache.org/documentation/tdb/

[25] Sparql query end point, available at http://dbpedia.org/sparql

[26] Apache Solr, available at http://lucene.apache.org/solr/