Bulletin of the *Transilvania* University of Braşov • Vol. 10 (59) No. 2 - 2017
Series I: Engineering Sciences

REAL-TIME DATA ANALYTICS WITH SEMANTIC WEB METADATA AND WEB SERVICES

G.A. STELEA[1] C. GAVRILĂ[1] V.P. FERNOAGĂ[1]

Abstract: *In modern web platforms, the new challenge is to provide a high degree of decentralization for the existing data and services. This paper is dedicated to Real-time Web analytics using standardized and accessible Semantic metadata and Web services to achieve improved performance in data visualization and increased competitiveness, which could open the way to the concept of real-time intelligence. We have shown how the usability, productivity and accuracy of exposed content in Internet can be increased by using Web Metadata and Semantics in order to define also the relationship between parts of the content. These methods, if put on the top of machine learning techniques, open new ways for creating "intelligent" web content.*

Key words: *real-time data analytics, semantic web, microformats, microdata, RDFa, cloud computing.*

1. Introduction

The Internet is a distributed system that allows developers to create applications, combining data [12] and services in original ways to produce new products and ideas [5].

In the current state of the Internet, in which the amount of data published is growing exponentially [13], tools that replace the human user with the task of processing and filtering information easily find their way in building new behaviour patterns [11] and make the first steps towards a world where people and machines become equal in the use of information.

The use of Semantic Web standards has real applicability to improve Internet browsing experience [6] by increasing efficiency, extracting and analysing information from user-free data feeds, assisting the user, analysing keywords for searches or other intelligence-driven activities. The development of web applications tends to integrate components quickly and with care over details that require less and less programming skills and are geared toward finding and using specific resources to provide a suite of unique functionalities [1].

In this paper, we propose a concept and a specific solution able to:

- determine the characteristics of Web users browsing and content preferences;
- collect data in order to improve the Web experience by dividing information into small, weakly connected parts;

[1] PhD student in "Electronics, Telecommunications and Information Technology", *Transilvania* University of Braşov.

- reduce complexity of applications and the integration effort by providing a set of reusable services, using standardized Semantic Web metadata and Web Services, in order to generate real-time analytics and process them - in an "intelligent" way.

2. Semantic Web Metadata

Semantic Web is an ongoing step in the evolution of Web architecture [8], allowing users to publish richer information in regular web pages that can encode machine-readable semantics [7], without relying on centralized services.

Standardization of Semantic Web metadata is in progress, some extensions are completely specified while others are in the intermediate stage [9]. The most popular three specifications of the Semantic Web are Microformats [10], Microdata and RDFa [14] (Resource Description Framework-in-attributes).

Regardless of the aspect, Semantic Web metadata are widespread either explicitly or implicitly by the semantics of the content and the structure of the marker, with the possibility of being explicit.

In this context, the Semantic Web initiative attempts to specify a standard for publishing information with the most common semantics, thus, marking the data from a semantic point of view, contributing to the evolution of the Web to a new stage where the information is equally accessible to people and machines [16], using single or multiple vocabulary sets for making semantic annotations in presentation languages (HTML).

In the proposed solution, the metadata specification used is Microformats and RDFa, as shown in Figure 1, and was designed primarily for people and secondly for machines. It uses a set of simple data formats built on existing and generally accepted standards, encapsulates semantic elements within HTML markups to facilitate decentralized development and represents conventions for names of HTML elements, attributes and values of attributes, with well-established semantics.

```
<div class="vcard" xmlns:foaf="http://xmlns.com/foaf/0.1/"
about="#me">
<img class="photo" property="foaf:image" src="profile.png"/>
 I am <span class="fn" property="foaf:name">George Stelea</span>
 from <span class="org" property="foaf:organization">
 Transilvania University</span>
 of   <address class="adr">
         <span class="locality">Brasov</span>,
         <span class="country-name">Romania</span>.
      </address><br>
  Website: <a class="url" rel="foaf:homepage"
  href="http://www.unitbv.ro/">www.unitbv.ro</a><br>
  E-mail: <a href="mailto:george.stelea@unitbv.ro"
  class="email v:email" rel="foaf:mbox">george.stelea@unitbv.ro
  </a><br>
  Phone:  <a class="tel" property="foaf:phone"
  href="tel:0040723123xxx">+40 723.123.xxx</a><br>
</div>
```

Fig. 1. *Microformats and RDFa markup in an HTML web page*

One of the most important features of this type of marking is the flexibility, the Microformats and RDFa markups are independent of each other, and are encapsulated in an HTML "div" element to describe the entity of an item. The markup is easy to process by a program that knows these conventions and the extracted data can be used for a variety of purposes by a tool that understands data semantics.

The markup was formatted with CSS methods, shown in Figure 2, without being influenced by the attributes and attribute values that specify the metadata, since they are just naming conventions accompanied by values specific to the particular context.

```
<style type="text/css">
  div.vcard { border:4px double #ddd; border-radius:15px;
  padding:10px 10px 28px 10px; width:400px; background:#f5f5f5;}
  img.photo {width:109px; float:left; margin-right: 8px;}
  span.fn {padding-top:9px; display:inline-block;}
  span.org {padding-top:9px;}
  address.adr {display:inline-block;}
  span.locality {font-style:italic;}
  span.country-name {font-style:italic;}
  a.url {padding-top:10px; display:inline-block; color:#10926e;}
  a.email {display:inline-block; color:#10926e;}
  a.tel {display:inline-block; color:#10926e;}
</style>
```

Fig. 2. *CSS rules used to format HTML markup*

Figure 3 presents the front-end view of the HTML block item, which is displayed in the web browser when a user visualizes and interacts with the web application.

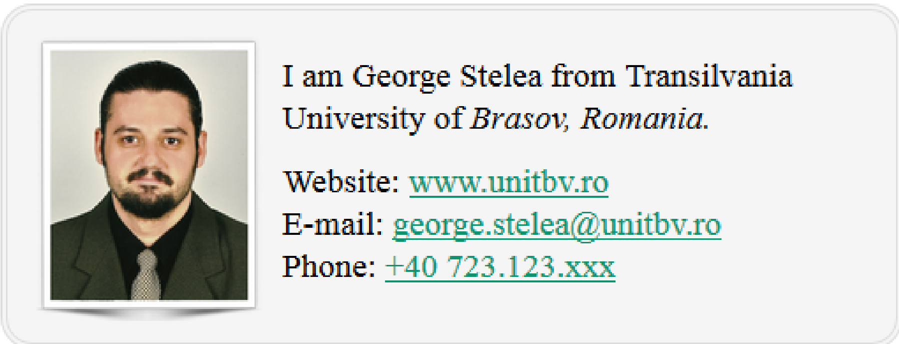

Fig. 3. *Front-end view of the HTML block item*

3. Web Services

The Internet is seen as a software platform in which users control their own data, making it usable at the disposal of others through collaborative tools and specialized services that can be used in a transparent manner by anyone.

A web service consists in a collection of functions packed and interpreted as a single, interoperable entity, published on the Internet for use by other programs. It involves the use of communication protocols that are platform, network and programming language independent and have associated processed data descriptions [17].

The communication protocol is meant to facilitate message traffic, providing a mechanism for structurally invoking and structuring data and support for extensibility, performance and reliability.

A parser was developed using JavaScript and PHP scripts to incrementally store the Microformats and RDFa metadata in a database which is then accessed by users in the web pages of the applications.

In order to support the transaction and data integrity, MySQL [4], an open-source relational database management system, was chosen for this solution, as presented in Figure 4. However, according to the requirements of the application, a NoSQL document oriented database for semi-structured data can also be used.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique More
2	name	text	latin1_swedish_ci		No	None		Change Drop Primary Unique More
3	organization	text	latin1_swedish_ci		No	None		Change Drop Primary Unique More
4	street	text	latin1_swedish_ci		No	None		Change Drop Primary Unique More
5	postalcode	varchar(45)	latin1_swedish_ci		No	None		Change Drop Primary Unique More
6	locality	text	latin1_swedish_ci		No	None		Change Drop Primary Unique More
7	country	text	latin1_swedish_ci		No	None		Change Drop Primary Unique More
8	website	text	latin1_swedish_ci		No	None		Change Drop Primary Unique More
9	email	text	latin1_swedish_ci		No	None		Change Drop Primary Unique More
10	phone	varchar(45)	latin1_swedish_ci		No	None		Change Drop Primary Unique More
11	profileImg	mediumtext	latin1_swedish_ci		No	None		Change Drop Primary Unique More
12	lastAccessedDate	varchar(45)	latin1_swedish_ci		No	None		Change Drop Primary Unique More
13	averageTime	varchar(45)	latin1_swedish_ci		No	None		Change Drop Primary Unique More
14	bounceRate	varchar(45)	latin1_swedish_ci		No	None		Change Drop Primary Unique More
15	numberOfInteractions	varchar(45)	latin1_swedish_ci		No	None		Change Drop Primary Unique More
16	activeReturning	varchar(45)	latin1_swedish_ci		No	None		Change Drop Primary Unique More

Fig. 4. *MySQL database structure*

The metadata (incrementally stored in the database) is exposed as a RESTful web service, where the result of a processing leads to the return of the representation of a Web resource.

The RESTful web services application development model was chosen because it requires a minimal infrastructure [3] which consists of implementing the HTTP protocol and the JSON open-standard file format, supported by most modern platforms and programming languages. For parallelization of the web services exposure, the XML format can be used in tandem with JSON.

The operations that can be applied to resources are GET, POST, PUT and DELTE.

The concept of RESTful web services is to use the existing and built-in standards without installing additional plugins and modules for data transmission, that can cause additional resource consumption and security leaks because of supplementary dependencies.

In Figure 5 a JSON resource representation is shown.

```
{   "id": 10,
    "name": "George Stelea",
    "profileImage": "profile.png",
    "organization": "Transilvania University",
    "address": {
                "street":"Bulevardul Eroilor 29",
                "postalCode":"500036",
                "locality":"Brasov",
                "country":"Romania" },
    "website":"www.unitbv.ro",
    "email":"george.stelea@unitbv.ro",
    "phone":"0040723123xxx" }
```


Fig. 5. *Representation of a resource using the JSON format*

The web services component is deployed independently of the application. This approach is preferred because it offers scalability and flexibility, so further changes or enhancements will not disrupt or affect the system. For enhanced results the information can be pre-processed locally (e.g. by some data-analysis algorithms of „edge computing”) prior to its sending to the real-time analysis in the Cloud.

4. Real-Time Analytics

In contrast to traditional analytics in which data is first stored and then processed, the concept of real-time analytics analyses data that is "still in motion", as soon as it enters the system [15]. In order to achieve real-time analytics in the proposed solution, the Microsoft Azure Stream Analytics service was used. Azure Stream Analytics (Figure 6) is a Cloud engine that allows real-time configuration and management and processes events and data flows from sensors, web applications, media feeds infrastructure systems etc. Using Azure Stream Analytics, large volumes of data can be examined using a structured decision-making process on a predefined logic, with immediate results being available in order to perform application tasks, generate alerts and automated workflows, perform simulations, modelling and optimizations based on a complete dataset and not just samples. The main advantages of Azure Stream Analytics is that it is cloud based, developer friendly and fully managed, with no upfront costs, no software/hardware maintenance, no performance tuning, that can be up and running in seconds and is only charged for usage.

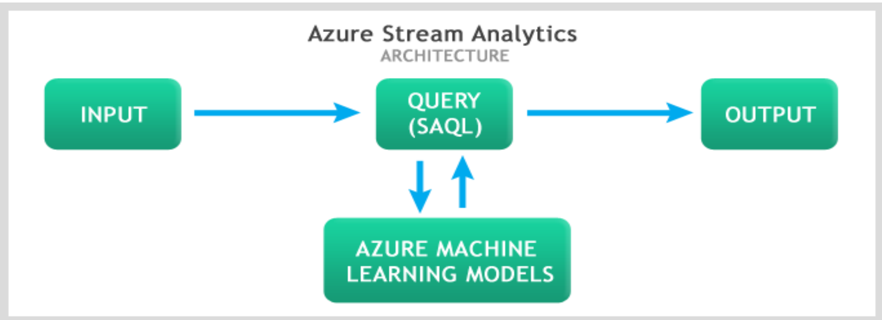

Fig. 6. *Microsoft Azure Stream Analytics architecture*

As input, the JSON objects were parsed using the Stream Analytics Query Language (SAQL), as shown in Figure 7.

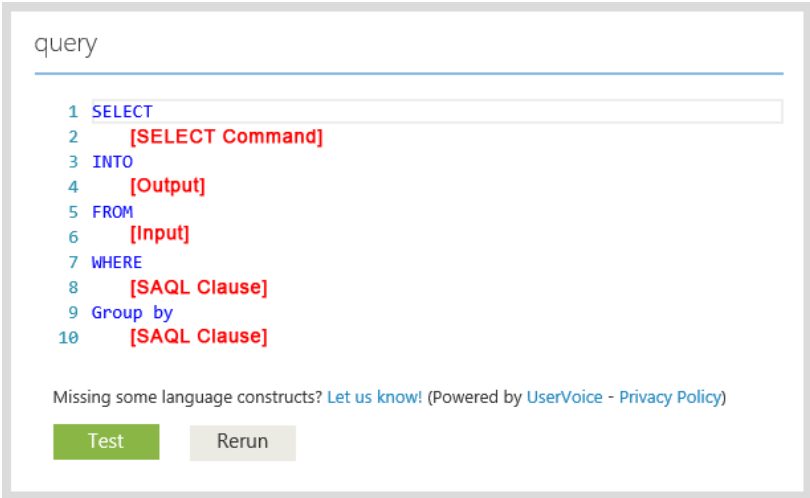

Fig. 7. *Queries in Azure Stream Analytics using SAQL*

The results are displayed in Microsoft Power BI analytics dashboard, as presented in Figure 8.

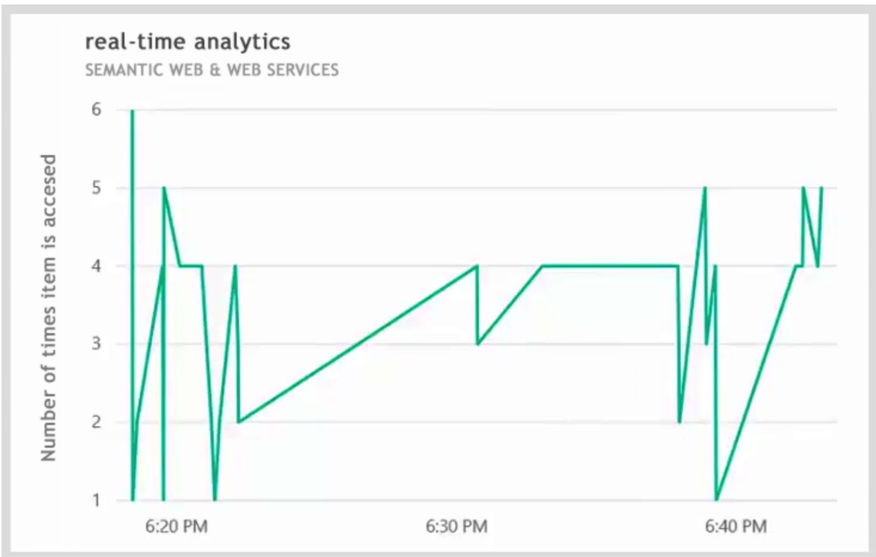

Fig. 8. *Microsoft Power BI analytics dashboard output*

One of the main goals of this solution being the simplicity of task implementation and portability, facilitating the process of management and administration, the development has been done in accordance with the SOA (Service Oriented Architecture) concept.

The Service Oriented Architecture requires the adoption of an application development style, considered as a service, which will be invoked by other applications. SOA supposes that new services can be created on the basis of existing ones, but the components of the system as a whole must have a high degree of independence [2].

Although Microsoft Azure Stream Analytics was used and presented, due to the independent mode of construction of the proposed solution which exposes the data as a web service, other real-time analysis solutions such as Amazon Kinesis Stream analytics and Software AG, or even custom developed solutions using decision tree classification, Bayesian classification or clustering methods can be implemented.

5. Conclusions and Further Development

In this paper, we highlighted a real-time web analytics solution and concept using transposable, reusable and standardized techniques that can validate service-oriented answers to data analysis problems, record information about users' behaviour on the Web and evaluating it without their feedback, as a continuous process running simultaneously with the navigation progress.

Standardized and accessible semantic metadata, both from the point of view of the publisher and the applications designed to process them, increase the semantic value of the web documents that contain them. Search engines are vital to achieve a successful user interaction experience. The lack of this way of navigating within applications would effectively diminish efficiency and dynamism.

Web applications require no installation, no space, no explicit maintenance, they can be accessed from the web browser on any device connected to the Internet, and because the solution is designed using independent and modular Semantic Web metadata and Web Services technologies, it has a broad practical applicability, unlike general and complex theoretical models that require adaptation and customization effort in order to be usable.

The concept of transformation, combining and enriching data is complemented by the ability to develop Application Programming Interfaces (API-s) and provide reusable data tools that can process JSON/XML formats, focusing on their presentation in the web browser, controlling the visual appearance of the application in order to provide a rich user interaction.

An API must be suitable for the audience to whom it is addressed, strong enough and clear enough to meet the needs of users and must be able to evolve over time. The API approach focuses on user experience thus the information mixing process becomes simple for the developer because of their ease of use and simple transfer formats, by spreading information through services instead of static pages. Also, AJAX technology can implement asynchronous Web Services that share data in XML or JSON format via HTTP protocol, in accordance to the client/server paradigm.

The importance of real-time analytics is increasing every year, and using it with machine learning can be the path of success for better efficiency in business insights, data visualization and increased competitiveness.

Web orientation will continue to be represented by the tendency for information exchange to be more open and transparent. Modern web applications must be based on open and not proprietary formats, because the Internet itself is a non-proprietary system that has to offer users transparency and the opportunity to collaborate and to exchange information online.

References

1. Buraga, S.: *Programarea in Web 2.0. (Programming in Web 2.0.).* Iași. Editura Polirom, 2007.
2. Duggan, D.: *Service-Oriented Architecture*. In: Enterprise Software Architecture and Design: Entities, Services, and Resources, Wiley Publishing (2012), p. 207-216.
3. Fielding, R.: *REST: Architectural Styles and the Design of Network-based Software Architectures*. In: Ph.D. Thesis, University of California, Irvine, USA, 2000.
4. Franke, C., Morin, S., et al.: *Distributed Semantic Web Data Management in HBase and MySQL Cluster*. In: Proceedings of the 4th IEEE International Conference on Cloud Computing (2011), p. 36-43.
5. Georgantas, N., Bouloukakis, G., et al.: *Service-Oriented Distributed Applications in the Future Internet: The Case for Interaction Paradigm Interoperability*. In: ESOCC 2013 - European Conference on Service-Oriented and Cloud Computing, Malaga, Spain, 2013, p. 134-148.
6. Gurparkash, K., Jaiteg, S., et al.: *Semantic Web Services in Clouds for Semantic Computing*. In: ACAI '11 Proceedings of the International Conference on Advances in Computing and Artificial Intelligence, Rajpura/Punjab, India, 2011, p. 229-232.
7. Hendler, J., Berners-Lee, T.: *From the Semantic Web to Social Machines: A Research Challenge for AI on the World Wide Web*. In: Artificial Intelligence **2** (2010) No. 174, p. 156-161.
8. Hitzler, P., van Harmelen, F.: *A Reasonable Semantic Web*. In: Semantic Web Journal **1** (2010), p. 39-44.
9. Hogan, A., Harth, A., et al.: *Searching and Browsing Linked Data with SWSE: The Semantic Web Search Engine*. In: Web Semantics: Science, Services and Agents on the World Wide Web **9** (2011), p. 365-401.
10. Loibl, W., Tüz, D.: *A Methodology for Building Microformats*. In: Information and Communication Technologies in Tourism, Springer, Vienna, 2012, p. 108-119.
11. Moreno-Vozmediano, R., Montero, R.S., et al.: *Key Challenges in Cloud Computing: Enabling the Future Internet of Services*. In: IEEE Internet Computing **17** (2013), p. 18-25.
12. Nishida, H., Nguyen, T.: *Optimal Client-Server Assignment for Internet Distributed Systems*. In: IEEE Transactions on Parallel and Distributed Systems **24** (2011), p. 565-575.
13. Pallis, G., Nguyen, T.: *Cloud Computing: The New Frontier of Internet Computing*. In: IEEE Internet Computing **14** (2010), p. 70-73.
14. Pereira, M., Martins, J.A.: *aRDF: A Plugin to Expose RDFa Semantic Information using Grails*. In: 6th Euro American Conference on Telematics and Information Systems (EATIS), Valencia, 2012, p. 1-8.
15. Steele, B., Chandler, J., et al.: *Real-time Analytics*. In: Algorithms for Data Science. Springer International Publishing, Switzerland, 2016, p. 381-401.
16. Tummarello, G., Mika, P.: *Web Semantics in the Clouds*. In: IEEE Intelligent Systems **23** (2008), p. 82-87.
17. Zheng, Z., Zhang, Y., et al.: *Distributed QoS Evaluation for Real-World Web Services*. In: IEEE 20th International Conference on Web Services (ICWS), Miami, Florida, USA, 2010, p. 83-90.