News Articles Platform: Semantic Tools and Services for Aggregating and Exploring News Articles

Koralia Papadokostaki
Technological Educational Institute of Crete
3 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Aris Papakonstantinou
Technological Educational Institute of Crete
4 PUBLICATIONS **1** CITATION
SEE PROFILE

Nikos Papadakis
Technological Educational Institute of Crete
45 PUBLICATIONS **127** CITATIONS
SEE PROFILE

Haridimos Kondylakis
Foundation for Research and Technology - Hellas
83 PUBLICATIONS **492** CITATIONS
SEE PROFILE

News Articles Platform: Semantic Tools and Services for Aggregating and Exploring News Articles

Koralia Papadokostaki[1], Stavros Charitakis[1], George Vavoulas[1], Stella Panou[1], Paraskevi Piperaki[1], Aris Papakonstantinou[1], Savvas Lemonakis[1], Anna Maridaki[1], Konstantinos Iatrou[1], Piotr Arent[1], Dawid Wiśniewski[1], Nikos Papadakis[1], Haridimos Kondylakis[2, a)]

[1]*Department of Informatics Engineering, Technological Educational Institute of Crete*
Heraklion, GR-71410, Greece
[2]*Institute of Computer Science, FORTH-ICS*
N. Plastira 100, V. Vouton, GR-70013, Heraklion, Greece

[a)]Corresponding author: kondylak@ics.forth.gr

Abstract. As the internet grows daily and millions of news articles are produced everyday worldwide by various sources, the need to store, index, search and explore news articles is more than prominent. In this paper we present an integrated platform dedicated to news articles, providing storage, indexing and searching functionalities, implemented using semantic web technologies and services. Besides using the developed APIs, the users through intuitive graphical user interfaces can save articles from RSS channels, import them through wrappers from external news sites or manually insert them using forms. A search engine on top allows the users to explore all registered information. All components have been implemented using semantic web technologies, using a novel ontology to model the news domain, a triple store for the management of data and web services exchanging JSON-DL messages. The registered articles become automatically part of the Linked Open Data cloud, enabling better data and knowledge sharing. Our preliminary evaluation shows the high-quality of the developed platform and the benefits of our approach.

INTRODUCTION

Internet is prevalent in our lives nowadays; access to it, is now easy and at high speeds and the availability of small portable devices has made it an integral part of our everyday life. It constitutes a basic medium for keeping ourselves updated. To this direction, following the news on the internet is gaining ground every day. Moreover, thousands of news items are produced everyday across the earth; whether important or trivial, they need to be saved, archived, searched and indexed. Information does not only involve pieces of text, but also images, video and audio files that complement the news articles and make their impact more significant. Lately, with the rise of social platforms, news articles can be shared through social networks, commented or even “followed” by external users.

In this exploding and demanding landscape, numerous research works focus on managing news articles. For modelling news articles for example BBC created its own ontology[1]. IPTC on the other hand, the global standards body of the news media, generated some widely-used mark-up languages[2] such as NewsML and rNews. In addition, the Schema.Org provides a well-known semantic vocabulary for modelling news articles[3]. Although those models are commonly used, a common data model integrating them all, linking them also with the emerging social media web

[1] http://www.bbc.co.uk/ontologies/storyline
[2] https://iptc.org/standards/
[3] https://schema.org/NewsArticle

sites is still missing. Besides modelling news articles, web-sites like NewsExplorer[4] and Event Registry[5] try to collect and annotate in real-time news articles. However, they provide limited APIs for importing and exporting articles, they do not expose the collected information as linked data limiting their potential and they miss linking the news articles with social media. Other approaches, assume that news articles are collected only through RSS feeds (Neptuno[6], NEWS [1] and Hermes[7]) or they are submitted via email (like myPlanet [2]) and others lack a central storage mechanism (using big files [3] to store articles) or miss a common data model (like SemNews [4]).

As a consequence, a new approach is required, unifying and linking the various data models and standards developed for modelling news articles, adding also the social dimension. In addition, the proper infrastructure should be in place offering flexible APIs and mechanisms for collecting, annotating, publishing, exploring and sharing of various news articles. To this direction, in this paper, we present *the News Article Platform* which aims to enable effective and efficient management of new articles using semantic web technologies and services. More specifically our contributions are the following:

- A novel ontology, named *News Articles Ontology*, to represent the required information and relationships among news articles integrating and linking all previous approaches adding also the social dimension.
- The necessary *APIs to enable pushing and pulling information* from a linked data repository publishing all relevant information as *Linked Open Data*.
- Besides programmatically interacting with the repository, our platform provides *an app with a friendly graphical user interface* (GUI) for *manually pushing news articles* and the relevant information to the repository and two easy ways to *automatically* insert new articles into it: i) via a *news wrapper tool* which allows the bulk insertion of articles from selected webpages; ii) via an *RSS crawler* which, given an RSS channel, allows the selective insertion of news articles into the repository.
- In addition, a search engine is provided, allowing the faceted search and exploration of all relevant information.
- A preliminary evaluation of the whole platform by 70 postgraduate students from the Department of Informatics Engineering at Technological Educational Institute of Crete shows the overall effectiveness of our platform and verifies the advantages of our approach.

To the best of our knowledge, the News Articles Platform is the only platform with an ontology representing all relevant information, and services to implement storing into and retrieval from a central repository, aggregators for the bulk insertion of data and a search engine. After the insertion of the articles into the repository, all data are made available as Linked Open Data, enabling their effective and efficient sharing and reuse. The rest of the paper is structured as follows: Section 2 elaborates on the architecture and the different components of our platform and Section 3 presents the preliminary evaluation performed. Finally, Section 4 concludes this paper and presents directions for future work.

ARCHITECTURE

A three-tier architecture was used for the implementation of the platform. It is illustrated in Figure 1 (a) and is consisted of the Data Layer, the Service Layer and the GUI:

- **The Data Layer**: In the data layer, which is the lowest in the architecture, a Virtuoso[8] triple store is used to store the ontology and the collected data. Virtuoso is a multiprotocol server, enabling data integration, publishing linked data and managing RDF databases. Virtuoso offers an ideal solution for the data management layer of our project as it can handle massive RDF data and it is free and open-source. Although the use of GraphDB (formerly known as OWLIM, which used to be open source) was at first considered as a candidate solution for data management, the fact that it is now a commercial product played an important role adopting Virtuoso.

  In order to model all relevant information in the domain, we created an RDF/S ontology named *News Articles Ontology*. RDF is among the most widely used standards for publishing and representing data on the

[4] http://emm.newsexplorer.eu/NewsExplorer/home/en/latest.html
[5] http://eventregistry.org/
[6] http://ir.ii.uam.es/neptuno/
[7] http://hermesportal.sourceforge.net/
[8] http://virtuoso.openlinksw.com/

Web [5]. The representation of knowledge in RDF is based on triples of the form (*subject*, *predicate*, *object*) whereas RDF datasets have attached semantics through RDFS[9], a vocabulary description language. The main purpose of the designed ontology is to be used as the core data model for creating, publishing, storing, indexing, and retrieving news articles integrating all previous models and approaches. To generate the ontology we carefully considered a) the fundamental classes exemplified in Schema.org, b) associations described in FOAF[10] c) the BBC ontology and c) the schema of the NewsML and rNews mark-up languages. As such we reused the core classes specified in past approaches such as "*Article", "Author", "Type"* and *"Comment* and their corresponding properties identifying their equivalences with the Schema.Org, the FOAF and the BBC ontology (for reasons of readability in Figure 1 (b) we only demonstrate equivalences with the first two ontologies). In addition, among other extensions and additions, we added classes and properties modelling social media and now an author might be a member of a social media web site and an article can be "shared", "liked" or "commented" using one or multiple social media accounts (Facebook, Twitter, LinkedIn, Google+).

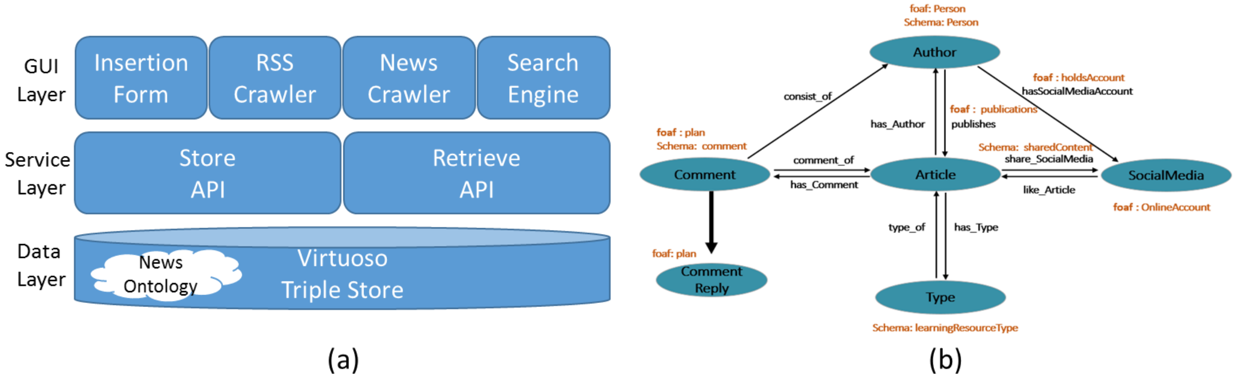


FIGURE 1. The three-tier architecture of our platform (a) and the core classes of the News Articles Ontology and their equivalences with Schema.org and FOAF (b).

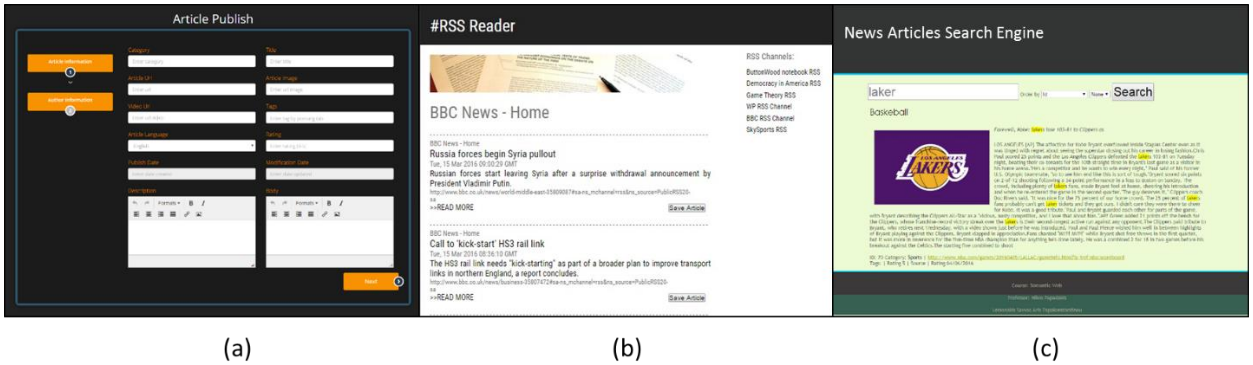
- **The Service Layer**: The service layer consists of two individual APIs one for storing information and one for retrieving. Both the APIs are developed as web services and adhere to the REST architectural design with constraints, namely RESTful APIs. Currently only the Retrieve API is publicly available.

  The Store API: The Store API is a set of web services. Those services receive requests from any application for data insertion, they create and execute the corresponding SPARQL queries to the Virtuoso triple store and they return the proper messages according to the server response. SPARQL[11] is the official W3C Recommendation for querying linked data. Most forms of SPARQL queries contain a set of triple patterns called a basic graph pattern. Triple patterns are like RDF triples except that each of the subject, predicate and object may be a variable. A basic graph pattern matches a subgraph of the RDF data when RDF terms from that subgraph may be substituted for the variables. Besides SELECT SPARQL queries for selecting data INSERT SPARQL queries are used to add triples and DELETE SPARQL queries to delete specific triples from the triple store. The implemented API has the capability to check for duplicates returning the proper warning messages. The Store API is developed using PHP, accepts http POST and GET requests and communicates with the Virtuoso triple store using SPARQL queries using a PHP library, named SPARQLib. All web applications developed on top of the service layer such as the web tool for the manual insertion, the RSS crawler and the news wrapper use the corresponding functions from the store API to achieve data storage.

  The Retrieve API: The Retrieve API on the other hand is a set of web services that receive requests for data retrieval, create and execute the proper SELECT SPARQL queries to the Virtuoso triple store and return the results in JSON-LD format. JSON (JavaScript Object Notation) is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. It is the most common data format used for asynchronous browser/server communication, largely replacing XML which is used by AJAX. JSON-LD is a lightweight Linked Data format. Linked Data empowers people that publish and use information on

[9] https://www.w3.org/TR/rdf-schema/
[10] http://xmlns.com/foaf/spec.
[11] https://www.w3.org/TR/rdf-sparql-query/

the Web and is a way to create a network of standards-based, machine-readable data across Web sites. It is easy for humans to read and write, based on JSON format and provides a way to help JSON data interoperate at Web-scale[12].



BBC News - Home

BBC News - Home
Russia forces begin Syria pullout
Tue, 15 Mar 2016 09:00:29 GMT
Russian forces start leaving Syria after a surprise withdrawal announcement by President Vladimir Putin.
http://www.bbc.co.uk/news/world-middle-east-35809087#sa-ns_mchannel=rss&ns_source=PublicRSS20-sa
>>READ MORE
Save Article

BBC News - Home
Call to 'kick-start' HS3 rail link
Tue, 15 Mar 2016 08:36:10 GMT
The HS3 rail link needs "kick-starting" as part of a broader plan to improve transport links in northern England, a report concludes.
http://www.bbc.co.uk/news/business-35807472#sa-ns_mchannel=rss&ns_source=PublicRSS20-sa
>>READ MORE
Save Article

laker
Order by: Id
None
Basketball

FIGURE 2. The web app for the manual insertion of articles (a) the RSS reader (b) and the search engine (c).

- **The GUI Layer**: This layer contains a *search engine*, a *web app for the manual insertion* of articles, authors and all relevant information, an *RSS crawler* and a *news wrapper*. All applications have been developed using HTML5, Bootstrap, JQuery and PHP in order to be user-friendly and up to date, and they rely on the APIs provided by the service layer. A short description of each app is provided below.

  *The web app for the manual insertion of articles***:** The web tool for the manual insertion of the articles comprises of an interactive, user-friendly form, which implements the insertion of articles into two steps. It contains the most important fields of the ontology and is also equipped with validation rules, thus allowing the error-free insertion of the corresponding information. A screenshot of the interface is shown in Figure 2(a). Using the form, articles, authors, comments, rating, images, videos etc. can be inserted whereas the inserted information is automatically linked to relevant information already existing in the Virtuoso triple store. Additionally, the insertion of duplicate articles or authors is not allowed. The web tool posts the relevant information to the proper API calls and visualizes the returned messages.

  The RSS crawler: Besides manually inserting news articles we have also implemented an RSS crawler which crawls existing RSS feeds, retrieves the corresponding news items and uses the API for inserting and storing the extracted information to our repository. RSS (Rich Site Summary) is a format for delivering regularly changing web content. Many news-related sites, weblogs and other online publishers syndicate their content as an RSS Feed to whoever wants it. RSS provides a standard easy method to retrieve the latest content from selected sites. The number of sites offering RSS feeds is growing rapidly and includes big names like Yahoo News. As such, the RSS crawler reads and displays initially one or multiple RSS channels and allows the selection of news items to be further saved into our database. As soon as an article is selected, the relevant parts are exported, and the corresponding calls to the Store API are issued. The interface of the RSS crawler is as simple as possible and a screenshot is shown in Figure 2(b).

  The news wrapper: The news wrapper is an app which allows the retrieval of news items from sites that do not offer RSS feeds. It uses the FiveFilters library[13] to parse articles out of news sites and blogs, and stores them in the Virtuoso triple store calling the appropriate Store API calls. Currently the generated app is called through the command line, getting as input a list of URLs, but soon a new updated version will be released with a fully-fledged web graphical user interface.

  *The search engine***:** The search engine of our platform is the app that allows searching into the Virtuoso triple store and visualizes the returned results. More specifically, users provide keywords and select the ordering of the results. The corresponding API calls are issued then using the Retrieve API. The Retrieve API accepts the request, formulates the correct SPARQL queries and returns the selected result to be visualized by the interface of the search engine. The results of the search can be ordered by various fields selected by the user interface whereas the faceted search is also possible searching in specific information types. For ranking

[12] http://json-ld.org/.
[13] http://fivefilters.org/

the results of the search engine a variation of the vector space model is implemented and used [6] calculating the cosine of the angle between the document vectors and the query vector – each vector contains the terms appearing in documents/queries. For the implementation of the search engine Html, JavaScript, PHP and CSS were used. The search engine receives a JSON-LD and shows the results of the query specified by the user. A screenshot of the first version of the interface is shown in Figure 2 (c).

PRELIMINARY EVALUATION

Deploying a platform which meets our initial objectives along with user satisfaction, ease of use and absence of defects was our target throughout the whole life cycle of our software. In order to achieve it, norms such as the Software Product Quality Requirements and evaluation (SQUARE) [7], defined from the International Organization for Standardization (ISO), have been used as a reference model As a result, the applicable functional and non-functional requirements according to ISO/IEC 25023 [8] had been defined in the early stages of the project and monitored by the developers during the whole software life-cycle.

For the evaluation, the quality features from the product quality model of the ISO/IEC 25000 series together with the System Usability Scale (SUS) [9] for global assessment of systems usability were used. At the evaluation stage 70 undergraduate students of the Department of Informatics Engineering, Technological Educational Institute of Crete were asked to complete a questionnaire with simple, accurate, non-time consuming, set of questions.

TABLE 1. The results for the various evaluation categories.

Functionality	Suitability	3.98	4.01
	Accurateness	3.83	
	Compliance	4.21	
Efficiency	Time Behaviour	4.27	4.14
	Resource utilization	4.00	
Compatibility	Co-existence	4.19	4.17
	Interoperability	4.15	
Usability	Understandability	3.75	3.94
	Learnability	4.04	
	Operability	4.02	
	Attractiveness	3.96	
Reliability	Fault tolerance	3.71	3.70
	Recoverability	3.69	
Maintainability	Analyzability	3.92	3.99
	Changeability	4.13	
	Stability	3.94	
	Testability	4.00	
Portability	Adaptability	4.00	3.99
	Installability	4.00	
	Conformance	3.94	
	Replaceability	4.04	
Quality of use	Effectiveness	3.83	3,86
	Efficiency	3.85	
	Satisfaction	3.90	
SUS	79.18 / 100		

Still, the questions had to be without loss of functionality/quality, so the crucial sub-features of software quality measures from ISO/IEC 25000 series have been formed into simple questions in natural language. The evaluation form of the platform was a set of questions where the evaluator had to answer within a degree of satisfaction using Likert scale [10].

The results of our evaluation are illustrated in Table 1. Values greater than three represent high level of the specific software feature; values between 2.5 and 3 are at low risk, whereas values below 2.5 are considered of high risk. In our case, all were graded with a satisfactory average above 3.7 showing the high quality of the software. The lowest average score was for the reliability (3.7/5) since in some cases, errors appeared when inserting documents without allowing a proper recovery. User feedback was used to correct all errors. The highest average score for compatibility (4.27/5) shows that the system adopts state of the art interoperability standards, and modules can be replaced at will without affecting the behavior of the entire system. When it comes to functionality, efficiency and compatibility our system receives high values, thus confirming its quality and high performance.

The SUS usability score was 79.18 on a scale of 0 to 100. The average SUS score from published studies has been measured by Sauro et al. [11] as the 62.1 but as a gold standard is often used the 68. Therefore, SUS score exceeding 68 are considered above average, while SUS scores below 68 are below average. Our SUS score of 79.18 exceeds by far the reference point of 68, yet future improvements can be made to provide even higher levels of understandability and fault tolerance.

The evaluation provided us with valuable feedback on the current state and with suggestions towards its improvement. For instance, minor problems regarding the search engine were located and were improved.

CONCLUSION

This paper presents a novel platform for aggregating, indexing and searching news articles as Linked Open Data. The platform consists of web service APIs to allow seamless integration of the GUI with the repository, an open source repository with SPARQL capabilities allowing the data to be stored as instances of our ontology. Moreover, an RSS crawler allows the uninterrupted insertion of articles and a news wrapper automatically parses articles from sites that do not expose RSS capabilities. The manual insertion of authors and articles is also possible through a user-friendly GUI. Finally, we enable effective and efficient search to the contents of our repository through a powerful search engine.

Currently we are experimenting with NLP solutions to understand the content of the articles and to be able to provide not only keyword search but semantically-enabled answers to user questions. In addition, we plan to release soon all tools and services in an open source repository to be publicly available as well.

In an era where the information is generated at tremendous rates, platforms for effective and efficient aggregating and searching news articles are of utmost importance.

REFERENCES

1. N. Fernández, D. Fuentes, L. Sánchez and J. A. Fisteus, “The NEWS ontology: Design and applications”. Expert Systems with Applications, 37(12):8694-8704 (2010).
2. Y. Kalfoglou, J. Domingue, M. Enrico, M. Vargas-Vera and S. Buckingham-Shum, “MyPlanet: an ontology-driven Web-based personalised news service”, IJCAI Workshop on Ontologies and Information Sharing, Seatle, WA, USA (2001).
3. M. Mohirta, A. S. Cernian, D. Carstoiu, A. M. Vladu, A. Olteanu and V. Sgarciu, ”A Semantic Web Based Scientific News” , 6th IEEE International Symposium on Applied Computational Intelligence and Informatics, pp. 285-289 (2011).
4. A. Java, T. Finin and S. Nirenburg, “SemNews: A Semantic News Framework”, in Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI), pp. 1939-1940 (2006).
5. G. Troullinou, G. Roussakis, H. Kondylakis, K. Stefanidis, G. Flouris, “Understanding Ontology Evolution Beyond Deltas”, EDBT/ICDT Workshops (2016).
6. H. Kondylakis, L. Koumakis, E. Kazantzaki, M. Chatzimina, M. Psaraki, K. Marias and M. Tsiknakis, “Patient Empowerment through Personal Medical Recommendations”, MEDINFO 216,(1117) (2015).
7. ISO/IEC 42010:2007, Systems and software engineering -- Recommended practice for architectural description of software-intensive systems (2007).
8. ISO/IEC DIS 25023, Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Measurement of system and software product quality.
9. J. Brooke, “SUS-A quick and dirty usability scale, Usability evaluation in industry”, 189(194):4-7 (1996).
10. R. Likert, “A Technique for the Measurement of Attitudes”, Archives of Psychology, 140:1–55 (1932).
11. J. R. L. Sauro, “Correlations among prototypical usability metrics: evidence for the construct of usability”, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1609-1618. ACM (2009).