Designing an Ontology-based Zika Virus news authoring environment for the Semantic Web

Edgard Costa Oliveira
University of Brasília, Software Engineering Faculty UnB Gama
Brasília – Brazil
5561992196093
ecosta@unb.br

Edison Ishikawa
Universidade de Brasília
Dept. of Computer Science
Brasília. Brasil
5561 993928784
ishikawa@unb.br

Lucas Hiroshi Hironouchi
University of Brasília, Software Engineering Faculty UnB Gama
Brasília – Brazil
5561 99903-1143
lucashh@hotmail.com

Thabata Hellen Granja
University of Brasília, Software Engineering Faculty UnB Gama
Brasília – Brazil
5561 98197-7034
thabata.helen@gmail.com

Marcos V. de A. Nunes
Universidade de Brasília
Dept. of Computer Science
Brasília. Brasil
5561 98651-5221
marcosnunesmbs@gmail.com

Daniel Rodriguez
University of Alcalá
Dept of Computer Science
Madrid - Spain
34 918856933
daniel.rodriguezg@uah.es

Rafael Batista Menegassi
University of Brasília
Dept. Computer Science
Brasília – Brazil
5561 992196093
batista7r@gmail.com

Luciano Gois
Brasilia Heath Department
SES-DF, Suplans
Brasília – Brazil
5561 99981-9081
luciano.gois@saude.df.gov.br

George Ghinea
Brunel University London
Dept. of Computer Science UK
& Faculty of Technology, Westerdals Oslo School of Arts, Communication and Technology,Norway
44(0)1895266033
george.ghinea@brunel.ac.uk

ABSTRACT

This paper describes the experience of researching and teaching the conceptual and practical basis for the specification, modelling and design of an ontology-based news authoring environment for the Semantic Web, that takes into account the construction and use of an ontology of the Zika disease. It has been said that CMSs are being adapted in order to receive semantic features, such as automatic generations of keywords, semantic annotation and tagging, content reviewing etc. We present here the infrastructure designed to foster research on semantic CMSs as well as semantic web technologies that can be integrated into an ontology-based news authoring environment.

Categories and Subject Descriptors

D.3.3 [**Software and its engineering- Software notations and tools**]: - Formal language definitions – *Semantics.*

General Terms

Semantic-based environment; Requirement specifications, Authoring environment architecture;

Keywords

Semantic Web, Ontology, Authoring tool, Content Management System – CMS, Semantic authoring.

INTRODUCTION

Nowadays, text authoring can be seen as a similar practice to those taken 100 years ago, with a slight difference: we have shifted from the hand-pen-paper model in cellulose (that still exists), to the digital finger-keyboard-cursor-white page. In the support level a lot has changed - such as making links to other documents; making and sending as many copies as desired - as we can see from the development of editing resources, which were in the past restricted to editing houses and their complex software. In the syntactic level, we can benefit from searching and ordering key words. However, in the semantic level, text production is the same as before: it depends on the writer´s ability to associate his contents to existing formal concepts structures (links with other documents, links to web pages, associating text to dictionaries, terminologies, taxonomies, indexes, etc).

In the Semantic Web, we are facing a new opportunity to use concept referencingability of a text – and not only its objects and components such as summaries, images, links, descriptive terms and their meanings. The main problem we are facing today is that the available content on the Web is generated by one person, indexed by another and retrieved by computers that do not make a difference between variant terms.

Based on previous studies [1], we have defined an ontology-based authoring environment for the Semantic Web as "*a set of writing tools for writing, editing and representing documents that interactively support users (authors), allowing a better*

access and use of knowledge semantic representation during writing, by doing the following tasks: semantic annotation of documents, metadata creation, linking terms in the document with external ontologies; linking similar documents with each other, transforming citations in labelled links, etc."

Particularly, when it comes to preparing a journalistic text, users of CMSs – Content Management Systems – in newspapers newsrooms, they count with a blank screen to insert texts with basic formatting options that current editors offer. However, the problem is that these tools limit the use of correct terms, by not giving the author the awareness of using the best term to identify a certain subject as well as its variations. To identify the best keywords to label the subject, to produce tags that are semantically linked, other than hanging loose and ambiguous. This happens to be the case of the subject Zika, disease or virus. The impact of this problem is related to the news production: they may contain useful information but they were not well represented via keywords or hash tags. Our question is: can we use propose an ontology based CMS for the production of news articles, that is able to link a term with its classes or instances in the ontology?

1. OBJECTIVES AND GENERAL METHODOLOGIES OF THIS PROPOSAL

In this paper we intend to present the context of the creation of the solutions, its motivation and proposals, by indicating a semantically computational platform designed to receive the solution; the Zika Ontology construction process; a general model of the architecture and the support of the authoring environment via a semantic CMS.

This research started with a general specification of the environment, by using as a starting point the general requirement of ontology based authoring environments [2]. In this project, we had the collaboration of undergraduate and master students from the University of Brasília, and professors from Information Science, Computer Science and Software Engineering. The group worked under the program of the Advanced Topics Computer Course. We invited a group of local medical staff to join in the construction of a Zika ontology. The users of the solution are journalists from the Campus Online UnB´s newspaper, from the Communication Faculty. We divided the group in three parts: environments specification team; conceptual modelling and ontology construction team; requirements specification and software development process and engineering team.

In a nutshell, the proposal of this tool is to annotate terms and concepts used by writers/journalists and to relate these terms with the ontology of the subject, to create links with other information resources about that subject: existing news pages or any other page selected by the writer. Regarding the user's side, text can be produced by many journalists and go under different review processes, but tagging and terminology consistency will be provided and supported by the ontology that will guide users in the task of choosing a term to be used and making the links between this term and its related concepts (synonyms, hyperonyms, antonyms, related subjects,etc).

2. DESCRIPTION OF THE REPRESENTATION LANGUAGES USED

Since the start of this course, due to the teams know-how, we decided to use Python to manipulate RDF models. Considering that RDF was created to describe resources on the Web, the resource description framework is of great importance to help find a way to extract relevant resources. Therefore, we have decided to used RDF/XML due to its simplicity and as a first formalized serialization, according to [3], as a working model to represent the base ontology, initially constructed in OWL format. We intend to browse the structure of the ontology and to recover classes and instances more relevant to the specific contexts about which are being written. Also by identifying the relations between classes and instances that were listed. Python with RDFlib [4] worked well for the initial development of the application. Schiessl [5] reveals that the RDFlib library is easy to use via parsers and serializers of RDF/XML data and is best used in small projects. We proposed, thus, to use RDFlib to deal with RDF and OWL data in a Python environment and SPARQL implementations.

Python is a small scale language [4] recommended for optimized performance and has simple implementation characteristics. We learned that Apache Jena Fuseki version 1.1.0 [6] was used to overcome the low performance verified by RDFlib. The free and open source solution based in Java is largely used to the development of Semantic Web applications. It showed efficiency in storing RDF data, good interface to submit Sparql queries and good answer performances. Even though it does not use a specific API to connect to the Jena-Fuseki server, the problem was solved by using commands via operational system to reach the goals. We used Apache Jena, a large-scale Java platform, designed for optimized performance. Indexation takes place via semantic annotation. This process is necessary to unite and interlink documents in a semantic space defined by the domain ontology. NLP – Natural language processing – is the main used tool to identify, compare and annotate documents. However, searching for minimizing possible effects of ambiguity, NLP was complemented by human validation.

The semantic annotation steps are as follows [5]: a) extract all ontological entities and lexical variations to a list; b) analyze documents and remove symbols and non-relevant text; c) analyze the text in order to extract relevant terms and lexeme; d) identify n-grams or other patterns; e) eliminate stopwords; f) compare with the ontology labels; g) indicate a grammar class to the term; h) indicate similarity of the term with the domain meaning; i) confirm the annotation via a domain specialist; j) add the annotation to the documents.

The RDFS, the RDF Schema [7] is a semantic extension of RDF and offers mechanisms to describe related groups of resources and the relationships between them. Daconta et al [8] present the main components of this vocabulary, described as follows: Classes: rdfs:Resources — it is the class of all other classes which are subclasses of this one;rdfs:Class — defines a group of related entities that share the same properties;rdfs:Literal — represents constant values such as texts and numbers; rdf:Property — defines a property of a class and the representing value;rdfs:domain — defines which class of a property it belongs to;rdfs:range — defines a group of possible values to a

property;rdf:type — a standard property to define an RDF subject in an RDF Schema; rdfs:subClassOf — specifies that a class is a specialization of another one; rdfs:subPropertyOf — declares that all resources that are related by a property are also related to other ones; rdfs:label — is an attribute that defines a label that is readable by humans.

To perform the search in the database, we used Sparql, which is a standard search language and a data access protocol. It means that Sparql allows more than access to the RDF triples – subject-predicate-object – or graphs but also to any data sources that can be mapped in the RDF. It allows the extraction of semi or fully structured data, to explore data used in unknown relationships queries, makes complex combinations of heterogeneous databases with simple queries, converts RDF data from a vocabulary to another and constructs new RDF graphs from queries in other vocabularies.

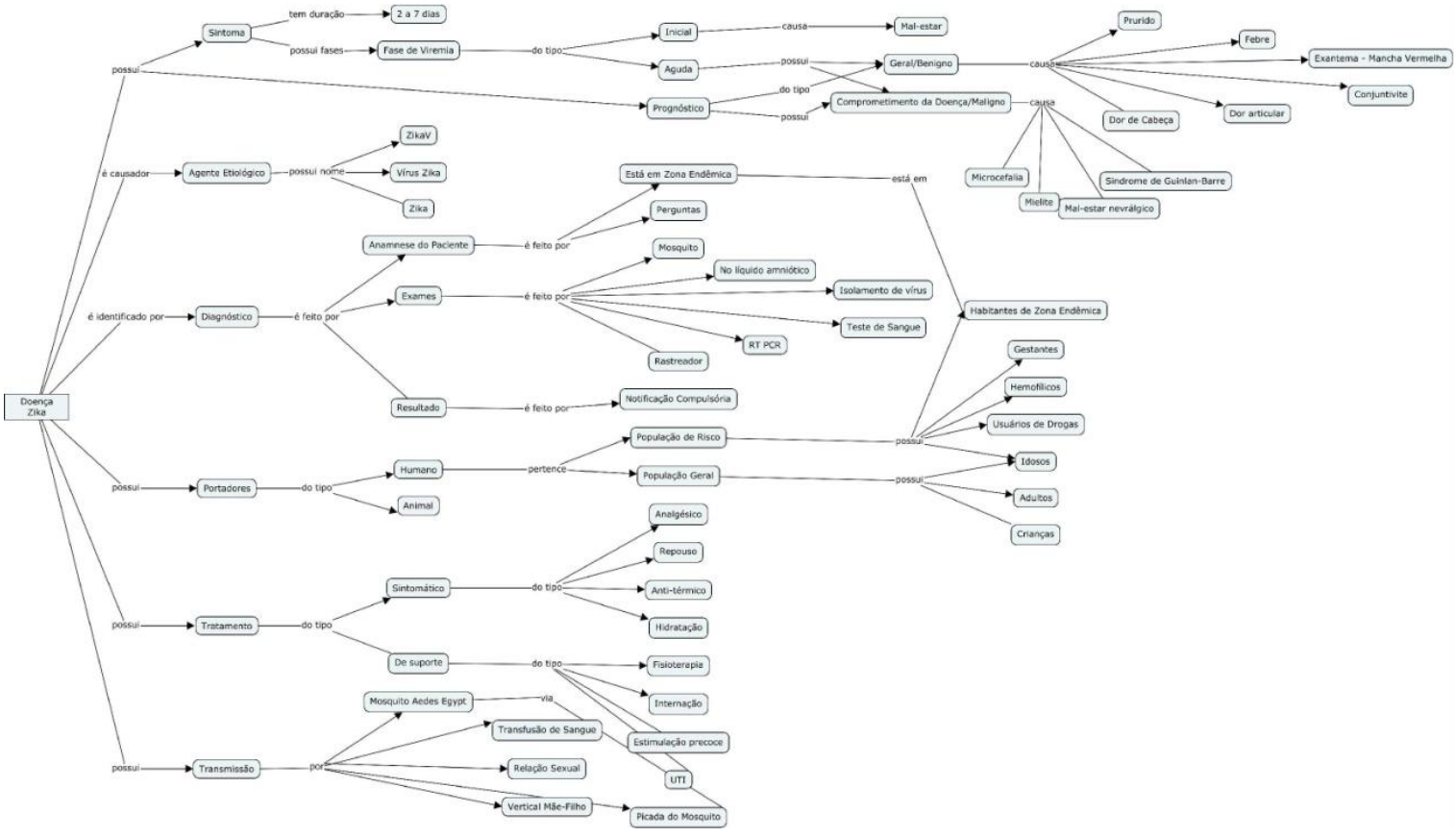
3. ONTOLOGY CONSTRUCTION METHODS USED

During the setting up of the authoring environment, a specific working group responsible for the ontology had meetings with a team of medical staff from the Health Department of Brasília, in order to generate a conceptual map of the Zika Disease (Figure 1). The meetings lasted 10h total approximately and the medical team informed all their knowledge about the virus and the situation of related diseased. The first artifact produced was the conceptual map for the understanding of the ontology domain. The results were homologated by the teams after the edition and there were some adjustments necessary for the construction of the ontology.

We used the 101 Methodology [9] to create the Zika ontology, from the University of Stanford, a simple method, whose authors also developed the ontology environment such asProtégé, Ontolingua e Chimaera [10]. This method is divided in phases: 1. Scope definition – from the meetings with the medical group, we defined the scope of the ontology.; 2. Consider reuse – there was no other ontology specifically about Zika, but some had Zika as an instance, however we based our research on the structure of these ontologies [11] and resource documents [12, 13, 14]; 3. Enumerate terms – all terms were numbered via XMind and then via CMapTools, from the meetings with medical staff as well as from searching the theme in specialized medical bibliography and news articles; 4. Define classes – this was complex and divergent, because deciding what is class or subclass can be confusing and time consuming. 5. Define properties – each class properties were identified in the conceptual map and were simple to implement in the ontology; 6 – Define restrictions – they were not used at this time due to scope and time limitations; 7 – Create instances – after we reviewed all classes and properties, we defined which were to become instances of the Zika ontology.

When considering reuse in the 101 Methodology, we identified that the term Zika is considered an instance within the ontology of diseases - Diseases (RDO:0000001) and Zika Virus Infection (RDO:0016040) – as presented in Figure 2.

Figure 1. Zika conceptual map



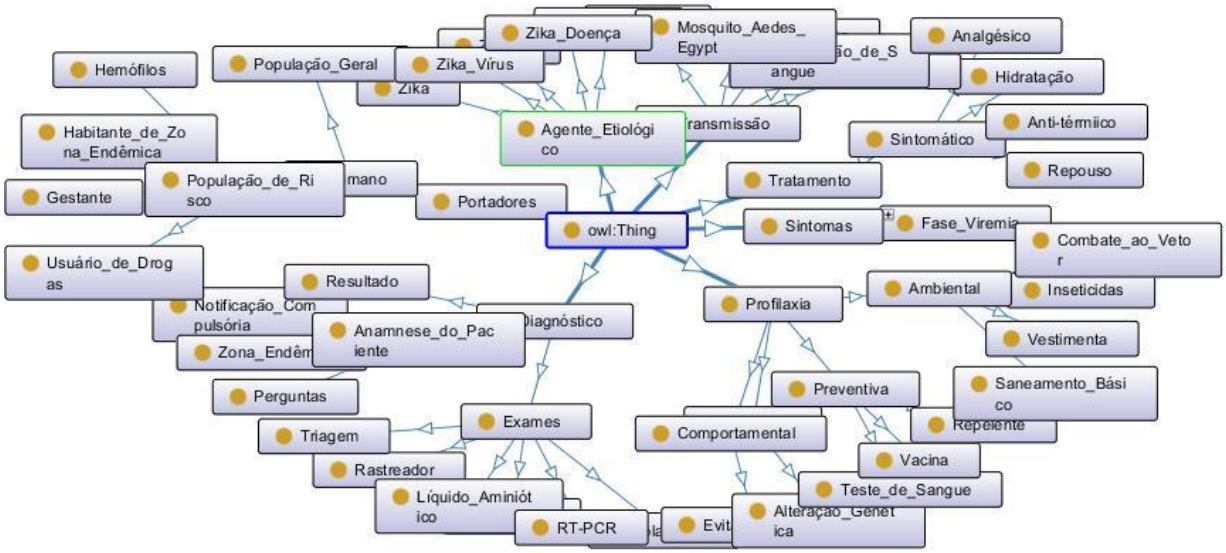


Figure 3. The Zika ontologyconstructed by the project´s team

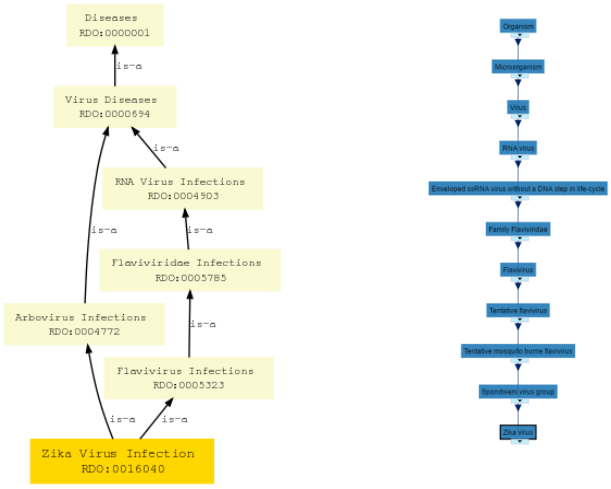


Figure 2. Ontologies that have Zika as an instance

The main definition of Zika was founded in RGD [15]: *A viral disease transmitted by the bite of AEDES mosquitoes infected with ZIKA VIRUS. Its mild DENGUE-like symptoms include fever, rash, headaches and ARTHRALGIA. The viral infection during pregnancy, however, may be associated with other neurological and autoimmune complications(e.g., GUILLAIN-BARRE SYNDROME; and MICROCEPHALY*).Zika virus (C0318793). In the Snomed CT [16] we found only a description of the Zika Virus as a final instance of a *Flavivirus* family. Thus we concluded the absence of a specific ontology about Zika, which allowed us to build a new one. After the definition of classes, relationships, properties and restrictions, we conducted the construction of the ontology itself with the support of Protegé.

Thus, after following all the steps and activities provided by the 101 Methodology, we generated the ontology, represented here by the graph on Figure 3 . The ontology project is at WebProtegé http://webprotege.stanford.edu/#Edit:projectId=6920c42c-cfc8-4d9c-8be7-c17a982a926e . This ontology was then validated by the group responsible for its construction, including the medical staff and other specialists in the subject area.We chose the reasoned Hermit to validate the Zika ontology to do its consistency checking capacities and to help review the general structure of the ontology.

4. THE SOFTWARE PROCESS DEFINED FOR THIS PROJECT

This project is an experiment to develop an ontology application in which undergraduate and master students learned the process and the contents via a PBL – Project Based Learning approach at the University of Brasília, UnB. Throughout the context of application done in the area of journalistic texts production about Zika, we needed to apply a flexible software development process along with the detailed documentation of all its guidelines.

For this purpose a unified and hybrid development process was defined, that was a combination of the RUP processes and the application of the Scrum methodology by using Kanban. From the traditional process we used general artifacts (architecture documents, glossary, iteration plan, vision document) and the phases (conception, preparation, construction and transition) were incorporated the Scrum methodologyinside the 5 sprints planed for the project. The vision document was the starting point to have a better notion of the scope of the product, however only in the architecture report the scope was finally defined and all the details designed and documented.

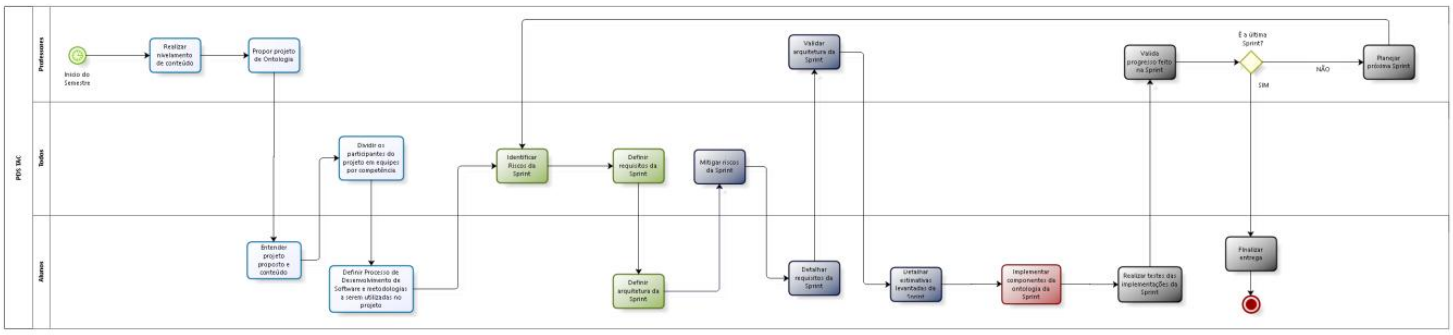


Figure 4. Software development process map created for the Project

The glossary was useful to a better understanding of the specific terms thatinvolves the context of the project. Finally, the iteration plan documented all the used methodology, as well as the division of the Sprints phases, as well as the definition of the platform used.

The iteration plan aims to give information of how the project organization was designed, including the integration of its activities, resources, deadlines, technologies and all the relevant information. A general scope, a selected development process and their guidelines were presentedas a result.

Considering that the project is about a new theme to the majority of the students, with the constant need to research the areas that involves our results, it was necessary to apply an agile method, thus, the Scrum methodology was chosen. There was also the need of another more precise documentation process and well-defined phases (Figure 4). For all these reasons and for the demand of an agile methodology used, the development process chosen for the project was a mixture of the traditional and the agile into a hybrid software development process.

As showed in Figure 4, the goal of each phase inside the sprint is arranged as follows: Initiation (or conception) with its focus on the understanding of the system; Elaboration: with its focus on the definition of the solution; Construction: with its focus on the implementation and to have the solution tested; and Transaction: with its focus on the implementation of the system in the context itself.

5. PROPOSED SOLUTION DESIGN

We searched for recommendations from theW3C about ontology driven applications as well as the ontology-based authoring environment previously designed [2]. Some of our main questions surrounded the following issues: RDF is a kind of a set of individual data saved in a schema based on an OWL graph. The retrieval of this data is made via Sparql, however there are some limitations: How can we write an RDF at each new register of an application? How can we modify this same RDF? Will it be necessary to create and RDF every time the new register is filled in? Is it possible to convert OWL to a structure model in SQL?

Then we defined some scenerios: today´s use of structural data bases, as showed in Figure 5 is a structured application that works in 3 main areas: logic and processing, data bases and visual. The user visualizes the screens, makes requests to the program and process the requests for data from the base, which returns the data that are interpreted by the application and shown to the user.

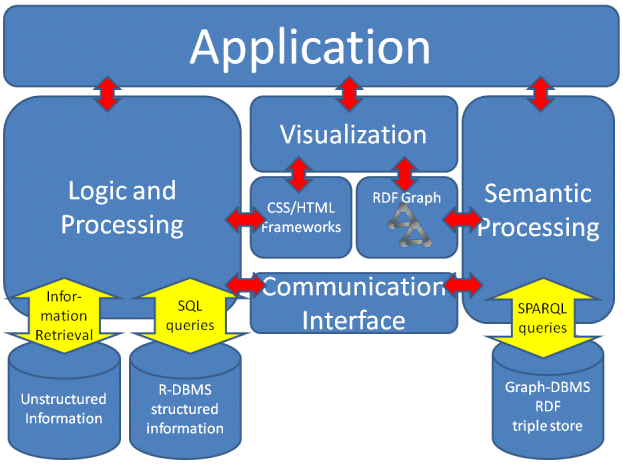


Figure 5. Models showing the use of the application via structural databases and with the use of ontology

We show a general view of a system architecturethat uses a series of different architectural views to illustrate the different aspects of the system. The intention is to capture and transmit the main decisions that were taken in regards to the system, from an architectural point of view.

5.1 PROTOTYPE PROPOSAL OF THE ONTOLOGY-BASED AUTHORING ENVIRONMENT

In a simplified manner, we present here how the project implementation was modeled, by showing each tool and language used. In order to implement the project, we found two viable paths: one using Python, according to the views of Jakub Talas et al [17] and another one based in the Java platform, as suggested by Isotani [10]. This research was conducted in an undergraduate course research, so we were free to try both paths.

The Python platform is composed of machines based in an Intel X86 Architecture, Linux Ubuntu O.S., Apache Server, applications in Python and libraries in Python RDFlib for the data treatment in the RDF and Django formats for the CMS.

Thus, the text authoring environment interface can present 3 modules: writing, ontology and semantic search engine: 1) we use Django to present a window of the document being edited, 2) another one with an RDF graph, corresponding to the semantic document annotation of the text being edited, and in a 3) third window the returned documents from the semantic search engine too that uses the RDF graph of the document to search for semantically related document.

The Java platform was set up with the following specifications. In a machine with and Ubuntu, we installed the Apache Tomcat software so that it is possible to manage a local sever based on Java servlets and supported by a Semantic CMS, here we suggested the Apache Stanbol [18]. In the issue of programming languages and supporting applications in the treatment of RDF files, we proposed the use of the applications Joint and Jena in Java, also counting on the support from Sesame/RDF4J in the handling of RDF files. Finally, information search and retrieval is based on Sparql, and this standard language used in semantic applications can be supported by the KAO implemented by Joint in order to refine the searches and retrievals made (figure 6).

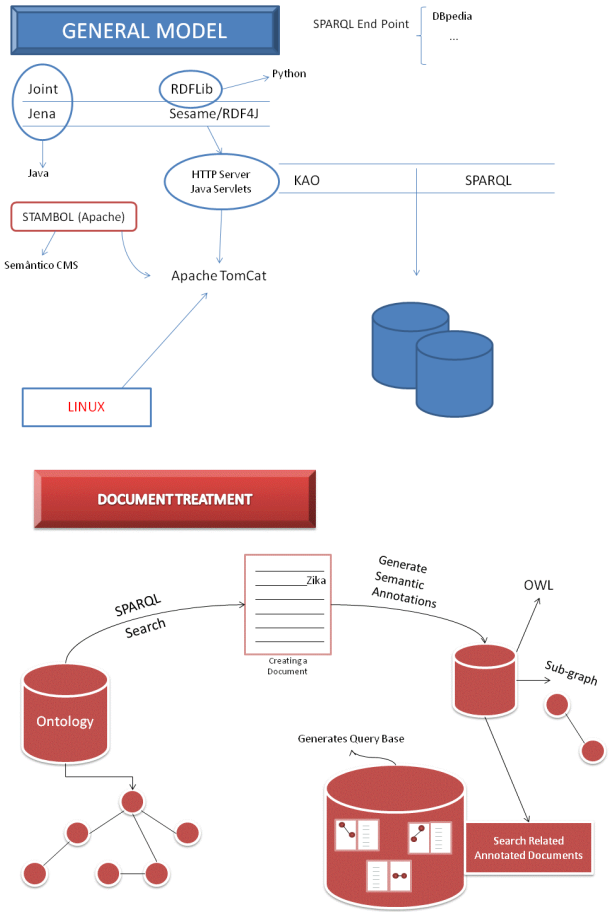


Figure 6. General model and document treatment view of the system architecture

Figure 6 shows a summary of the solution components model, where the application database comes from the domain ontology defined. While editing a text in a wiki (i.e. MediaWiki) environment, the platform recognizes the text edited via annotation with a Sesame RDF4J Joint and Jena application, and via servlets Java, interacts with the Apache Tomcat and Apache Stanbol [18], which is the solution for a semantic CMS. Classes and instances of the ontology are then matched with the text in order to support the annotation process. Considering the document treatment process, an XML Zika-subject text is being written, while annotations are made via connectionwith the OWL ontology, then making the semantic annotations and thus extracting a sub-graph with the ontology instances or classes that were recognized in the text. This sub-graph or sub-ontology is used to generate specific and context aware keywords and tags to represent the text, as well as to hyperlink it with other strictly related texts that are similar in concepts and terms.

6. CONCLUSIONS

We presented in this paper the experience taken in the Laboratory of Special Projects with students of the Department of Computer Science and Software Engineering of the University of Brasília with the challenge to understand and apply Semantic Web technologies to enhance the semantic capacity of CMSs. The main results of the experience, related in the paper, was the construction of an OWL Zika Ontology, the modelling of the authoring environment and the implementation of the database search mechanism.

The architecture model for a prototype of a semantic CMS was described and implemented in the lab. This model represents the effort and practice of the students who showed advanced abilities to deal with semantic web challenging issues in a computer science environment, even though these students were not familiar with these technologies before.

We brought here the following contributions to the area: the specification, modelling of an authoring environment for a text editor supported by a semantic and lexical interpreter for the edition of news articles about Zika, supported by a specific ontology created by the students. The following steps in this research is to implement the authoring environment in full, allowing real-time concept recognition from text annotation with the ontology.

This experience resulted in an environment that allows the use of a text editor, integrated to a semantic CMS, in which terms can be typed and in parallel be automatically recognized and associated to classes and instances of the Zika ontology. From the relationships created between the ontology and the text, one is able to obtain for this annotation a list of keywords and conceptual #tags that identify specific subjects of the text, the scope of the article in relation to the general context of the Zika ontology. It also correlates the text with already existing texts and articles or pages so that they can be interconnected via non ambiguous semantic relationships.

This work shows the feasibility in the use of ontologies during the moment of text production, that is, during the moment authors are deciding which terms to use in the text, in order to enhance information representation. The difference from other approaches is that the use ontologies mostly for post-publication or for information retrieval purposes. We also showed that it is possible to implement the solutions, not yet identified in existing CMSs available in the market which have not benefited from

ontology-based solutions that enhance knowledge representation capabilities.[i]

7. REFERENCES

[1] E. C. Oliveira. F.van Harmelen; M. Lima-Marques (2004). A framework for ontologyy-based authoring environments. In ISWC 2004 – International Semantic Web Conference, Hiroshima, Japão. 2004.

[2] E. C. Oliveira (2006). Autoria de documentos para a Web Semântica: um ambiente de produção de conhecimento baseado em ontologias. Universidade de Brasília, 2006 (Tese de Doutorado). 260p.

[3] F. Gandon,; G. Schreiber (2014). RDF 1.1 XML Syntax. Rio de Janeiro: W3C, 2014. Disponível em: http://www.w3.org/TR/rdf-syntax-grammar/. Acess in May 2016.

[4] RDFLIB. rdflib 4.2.2-dev. https://rdflib.readthedocs.io/en/latest/> Access in May 2016.

[5] M. Schiessl. Lexicalização de ontologias: o relacionamento entre conteúdo e significado no contexto da recuperação da informação. 2015. 261 f., il. Tese (Doutorado em Ciência da Informação) Universidade de Brasília, Brasília, 2015.

[6] Apache Jena. A free and open source Java framework for building Semantic Web and Linked Data applications. Configuring Fuseki http://jena.apache.org/index.html> Access in May 2016.

[7] W3C. RDF Schema 1.1 Recommendation 25 February 2014. http://www.w3.org/TR/rdf-schema/ .Access in April 2016.

[8] M. C. Daconta, L. J. Obrst, K. T. Smith. The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management. Willey, 2003.

[9] N. Noy, D. McGuinness, D. (2001) "Ontology Development 101: A Guide to Creating Your First Ontology," Stanford University: http://protege.stanford.edu/publications/ontology_development/ontology101-noymcguinness.html.

[10] S. Isotani e I. Bittencourt (2015) "Dados Abertos Conectados", Novatec Editora, São Paulo.

[11] CRRD. Ontology of Disease. Ontology browser of the CRRD. Bioinformatics Program, HMGC at the Medical College of Wisconsin. <http://crrd.mcw.edu/rgdweb /ontology/view.html?acc_id=RDO:0000001> Access in June 2016.

[12] P. Schram (2016) "Zika Virus and public health". J Hum Growth Dev. 26(1): 7-8. Doi: http://dx.doi.org/10.7322/jhgd.114415. Access in June 2016.

[13] L. Bushak (2016) "A Brief History of Zika Virus, From Its Discovery In The Zika Forest To The Global Outbreak Today", http://www.medicaldaily.com/zika-virus-outbreak-history-381132, Access in April 2016.

[14] A. Rasmussen, M.D., Denise J. Jamieson, M.D., M.P.H., Margaret A. Honein, Ph.D., M.P.H., and Lyle R. Petersen, M.D., M.P.H. (2016) "Zika Virus and Birth Defects — Reviewing the Evidence for Causality", http://www.nejm.org/doi/full/10.1056/NEJMsr1604338#t=article. Access in May 2016.

[15] RGD. Zikavirus infection. Gene Editing Rat Resource Center.http://rgd.mcw.edu/rgdweb/ontology/view.html?acc_id=RDO:0016040 Access in June 2016.

[16] BioportalSnomedCT. Zika Virus. http://purl.bioontology.org/ ontology /SNOMEDCT /50471002 Access in May 2016.

[17] J.Talas, T.Gregar, and T.Pitner (2011). Semantically Enriched Tools for the Knowledge Society: Case of Project Management and Presentation. In: Knowledge Management, Information Systems, E-Learning, and Sustainability Research Volume 111 of the series Communications in Computer and Information Science Springer, 2011.

[18] IKS. Developing Semantic CMS Applications: The IKS Handbook (2013). Editors WernherBehrendt and VioletaDamjanovic. Salzburg Research Forschungsgesellschaftm.b.H. 2013.

MEDES'16, 1-4 November 2016, Hendaye/France.