Original Research Article

An Adaptable Ontology for Easy and Efficient University Data Management at ~~in~~ Niger Delta University

.

ABSTRACT

The structure, variety and quantity of some current web content are ~~is~~ limited inefficient exploration due to difficulty in searching and locating ~~a~~ specific content. A case is the Niger Delta University where there is no unified online structure holding relevant university data. This is because there is no common data model to manage the data for which such query can easily be interpreted semantically. Therefore, this paper presents the application of semantic web technology for the unification of university data management. This research prepares the ground for the advent of software agents and other applications that require structured data for ~~its~~ their computational processes. The ontology development follows an iterative path of the Object-Oriented System Analysis and Design (OOSAD) methodology. SPARQL was used as the query language for testing the ontology. The result is semantically structured data that is deployable and ~~which~~ can be expanded ~~expended~~ and adapted in other institutions.

Keywords: Database, Ontology, Semantic Web, Sparql

1. INTRODUCTION

Semantic Web, which is also called the Web of Data extends the current web of documents. It is primarily aimed at extending the present web, which is made up of interlinked documents with hypertext and hypermedia to a web of linked data instead of documents. The intention is to enable machines (computers) to understand the content of the web, unlike the current web of documents whose content is for humans to understand only [11]. This web is an original idea of Tim Berners-Lee: the founder of the web [4,18], the idea was passed to the public stating its vision and goals [1].

The vast increase in the variety and quantity of the current web content has brought about some limitations to its efficient exploration in the sense that it is becoming more difficult to search for and locate specific content [13]. For example, it is problematic to search for either people with the same or common names, or famous people with the same names. This is due to the fact that all of the information may not reside in a particular location. As such, gathering and integrating the same from the various locations for an integrated aggregation and processing become challenging for this web [13]. The current web does not as well have the capacity to handle very complex queries as well as retrieval of information in a more general context that could be integrated, shared, and processed [8]. For example, a query to return all the names of EU countries heads of state can be very daunting and it seems to go beyond the present web query engine prowess. Even though the web has this required information readily available [19,7].

Here in Niger Delta University, student record is contained in three different portals hosted in three different server machines. The portals hold student academic records, biodata, and hostel allocations. Hence, a query to generate a specific Cumulative Grade Point Average (CGPA) of a particular student resident in a particular hostel in any of these portals will be difficult. The difficulty is with respect to the spread of information in the different portals. Since student hostel data is stored in student hostel management system and CGPAs are stored in student academic record management system which is hosted in a different server. Similarly, another related example is searching for the best student in a particular year and their indigeneity. Again, such a query will be problematic given that student academic record and biodata resides in different portals and servers.

This category of tasks typically requires information integration from more than one source. This kind of integration problems most of the time can always be solved using some sort of software “glue” to bring together the required information and or services from their varied sources or locations. This paper therefore is one geared towards the introduction of semantic web technologies in the Niger Delta University with the sole purpose of developing an ontology for students’ records in the University to allow for an integrated search processing and retrieval of required data and information. Also, we show how we can retrieve such information given an instantiated ontology, we applied SPARQL query language to extract relevant information.

2. LITERATURE REVIEW

The Internet and the development of the semantic web have created the opportunity to provide structured legal data on the web. The Semantic web can be defined as the set of technologies that extends the current web by way of enabling data connection directly from one document to the other on the web, for easy understanding and manipulation by both machines (computers) and humans [13,4]. This technology represents web content in a format that is understandable for both human and machines. It makes web content to be machine processable in which intelligent techniques can be applied to take advantage of these representations [1]. It is meant to interpret sentences and retrieve useful information for users. In light of this, some of the technologies that are necessary for achieving these functionalities include RDF, OWL AND SPARQL and so on.

An ontology describes an explicit conceptualization of a particular domain, which basically defines the model of the domain with possible restrictions [15]. Each group of single objects of classes defined with the use of ontology becomes a knowledgebase. The conceptualization only describes the knowledge derivable from the domain rather than the specific state of things within the domain [15]. The implication of this is that conceptualization does not change, or rarely change. Ontology is therefore a specification of this conceptualization. [16] defines ontological relationship existing between legal elements from information extracted from Thai criminal code. The ontology is built from concepts gathered from three components which are elements of crime, justification, and criminal impunity from Thai. The focus of the ontology is to facilitate users understanding and to provide interpretation in analyzing the legal elements of a criminal law. However, the focus of the ontology as well as its structural and hierarchical organization differs from an ontology for legal question answering. [2] reports an ontology and rules that capture and represent the relationship existing between the actors and their different roles in money laundering crime. Also, companies’ relational information is demonstrated in order to establish relationships: entities, people and actions.

Ontologies have been in increased used in different domains for the specification of a representational vocabulary shared in the various subject areas to improve search and integration as well as a resolution of ambiguities in concepts. Ontologies are the textual or graphical description of conceptualization of a particular domain. It provides different solutions to the semantic problems in understanding the legal subdomain, since they provide the building blocks for modelling legal concepts. Clearly defined ontology helps in solving ambiguity issue where a concept could mean different thing to one field or the other. Several researchers have applied ontological techniques in the different domains to solve reasoning problems and dealing with querying ontologies in natural language as well as information retrieval problems. One of such application is in [22], where ontological techniques was applied in the reasoning process for case-based reasoning. [9] applied ontology and rules for legal reasoning.

3. ANALYSIS OF THE EXISTING SYSTEM

Most Universities in Nigeria like the Niger the Delta University uses many portals for managing different aspects of their data, generated on daily bases. These data include Academic, Hostel, and Admission data. In the Niger Delta University, students' Bio and extra curriculum activities data is not well captured and managed. As a result, data retrieval of student is done in each of these isolated individual platforms even though the query require data from all the platforms. Which means that a query to retrieve such an information needs to be broken down into smaller queries that can run on individual platforms. The results are then integrated manually by the end-user to generate required information. Moreover, it is sometimes difficult to retrieve specific types of information since there is no common platform to send your query.

The data structure adopted in the portals is a relational database which is used in the storage and manipulation of data in the different platforms. These portals do not provide semantic information understandable by computers which makes it difficult for specialized queries. The current system makes the derivation of specific information very slow, and sometimes very difficult or impossible. This is because there is no correlation between the data in the platforms. What exists rather is a link between the documents in the different portals. Each application portal is tied to its own data (domain dependent). This means that the portals are developed to conform to the schema of the data structure, such that the data can be manipulated by the application. This makes it difficult for data sharing. Moreover, though the existing system is semi-automatic, however, it requires high human intervention for proper aggregation.

4. APPROACH

Broadly, our approach follows the Object-Oriented System Analysis and Design (OOSAD) methodology of the system analysis and design model. Our implementation consists of two tasks, (a) design an alternative data model using semantic web technology from existing structured relational databases and (b) the application of SPARQL query for retrieving semantic information. The encoding of the ontology is done by extracting relevant student information from the database schemas in the different portals to develop the student information management ontology. We identify specific information such as student, school, faculty, department, program and so on to create the ontology classes, superclass, subclass, object properties and data properties. However, our approach for constructing the student information management ontology follows three fine grain steps which are: enumeration of ontology terms, definition of ontology classes, and object properties, and creation of class and object property hierarchy.

Enumeration of ontology terms

Here, we manually write down all the database entities, relations, and properties. This step is very important and necessary in that it serves to enhance the identification and classification of relevant objects as classes and relations as well as properties for building the terminological base of the ontology. Moreover, database designers are fond of managing complex compound nouns while designing databases. For example, Student-name, Extra-curricular-activities, and so on. Also, other cases like lexically related terminologies are identified and their relatedness is checked using WordNet if there is a relevant path connecting them together. We used the Wu-Palmer path similarity measure to measure the similarity of such concepts [21]. The similarity scale used for the measurement is between 0.7 to 1.0 on a scale between 0 and 1. The extraction process applies a filtration technique to identify nouns and relations to generate a list of keywords useful for building a student record ontology.

Definition of ontology classes and object properties

From the list of keywords generated from section 4.1, we identified relevant keywords useful for our purpose. From the relevant keywords we determine the type of nouns and predicates which could be described as classes, instances, and object properties. For example, we identified nouns like *Person*, *Student*, *Lecturer*, *Student_number* and so on as classes while *Mary*, *student-numbers* and so on as instances of Person and Student_number classes respectively (See figure 1.0). Also, classes are categorized into atomic and definable classes. In same manner, amongst the relevant predicates we identify the predicates connecting the selected noun classes together. However, it is necessary to emphasize that it is not all keywords from the source database that are relevant in the construction of the student record ontology.

In same way amongst the identified predicates some are relevant, and some are irrelevant. The relevant ones identified are defined as the object properties. For example, we identify predicates like livesInCampus, hasLibraryRecord, hasBiodata, hasAcademicRecord and so on. These object properties were used to connect classes, superclasses and subclasses together.

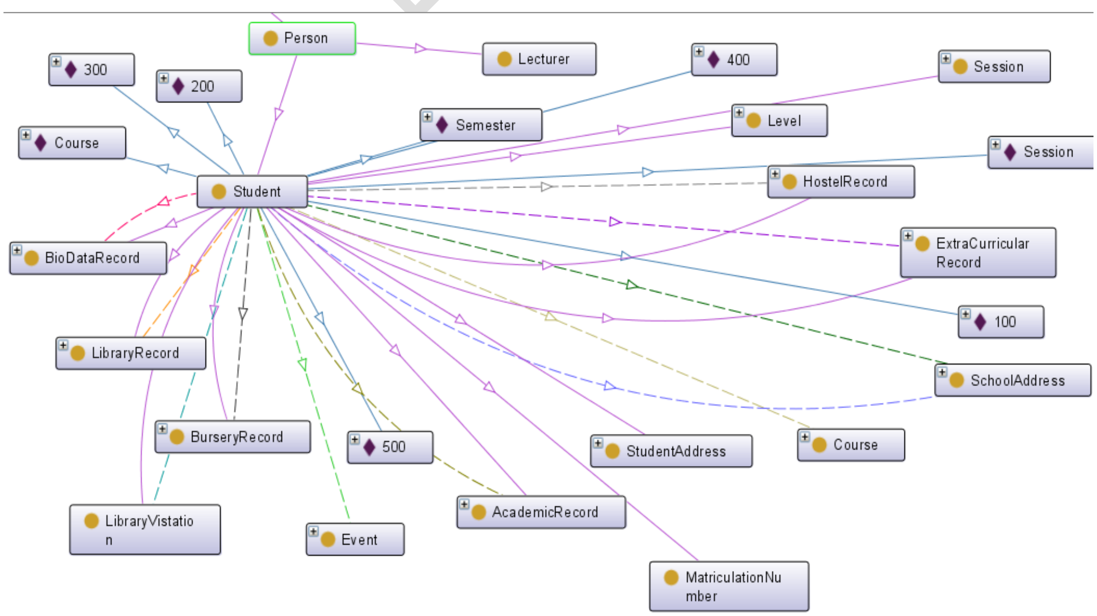

Figure 1. Student Ontology Summary

Creating class and object property hierarchy

For ontology classes and object properties hierarchy creation, we also identified other classes and properties which are not part of our source database but are useful in hierarchy creation as adopted from [10,14]. The essence is to enable us to create the respective class and object property hierarchy. For example, while creating the class hierarchy, the *Person* class is the superclass of *Student* and *Lecturer* class. Which means $Student \sqsubseteq Person$ and $Lecturer \sqsubseteq Person$. For class hierarchy creation in other classes like Faculty, Department and Program. The *Faculty* and *Program* classes are superclass and subclass of the Department class as $Faculty \sqsubseteq Department \sqsubseteq Program$.

We then define domains and ranges (i.e., classes) for the respective relations and characteristics identified as way a of creating restrictions on the relations. On the one hand object properties connect individuals from a particular domain to another individual from the range. For example, object property $takeCourse$ has Student class as the domain and Course as the range ($\exists takeCourse.\top \sqsubseteq Student$; $\exists takeCourse-.\top \sqsubseteq Course$); for $hasAcademicRecord$ has Person class as domain and Record as range ($\exists hasAcademicRecord.\top \sqsubseteq Student$; $\exists hasAcademicRecord-.\top \sqsubseteq Record$). The idea of property hierarchy is to relate the properties into superproperties and subproperties.

The Protege 5.2.0 ontology editor was used in constructing the student information management ontology in this research. The frame instances, slots and classes of Protege ontology editor corresponds to the individuals, properties, and classes of OWL ontology language. Classes are tangible expressions of concepts [5]. They are interpreted as a collection of individuals with similar structure. Pellet reasoner was used to check for consistency, classify taxonomy and compute inferred types. Pellet is a third-party open-source Java-based reasoner, used by protege 5.2.0 through a plugin. It contains novel optimizations for conjunctive query answering and incremental reasoning as well as nominals [20]. OWL2 and OWL2EL profiles supports Pellet reasoner. We also used the SPARQL query plugin to construct SPARQL queries. RDF Library and Ontograf plugin were used for Ontology visualization.

5. SPARQL QUERY LANGUAGE

The W3C accepted language for query knowledge graph is the Sparql query language. It provides an interface to the knowledge graph which allows for the querying of classes and subclasses of an ontology. It applies the graph matching technique to match patterns (triple patterns) in the graph to retrieve information from a knowledge graph [3,17]. It allows users to state the triple patterns based on *subject*, *predicate*, and *object* which is then executed in an RDF dataset. To ascertain the success of the ontology development process, the sparql query language was used to retrieve semantic information from the ontology. A semantic interpretation is viewed to be correct if the system retrieves the precise answer based on the sparql query presented. Sparql queries grant data users the enablement to integrate distributed data across the web. With the sparql language, users can query a knowledge base using the sparql endpoint. Queries are directed to a particular sparql endpoint, and the results are received from the federated query processor [6,12]. To measure the quality of the ontology, we populated the ontology with respective information and executed some sparql queries to retrieve some specific information. See figure 2, 3,and 4 for some selected sparql queries and the respective outputs.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ndu: <http://www.semanticweb.org/thompson/ontologies/2018/2/NDUStudent#>
SELECT ?StudentFirstName ?StudentLastName  ?Student ?CourseCode
        WHERE {
                ?Student ndu:hasAcademicRecord ?AcadRecord.
                ?AcadRecord ndu:takesCourse ?CourseCode.
                ?Student ndu:StudentFirstName ?StudentFirstName.
                ?Student ndu:StudentLastName ?StudentLastName.

              }
```

StudentFirstName	StudentLastName	Student	CourseCode
"Iniakpokeikiye"^^<http://www.w3.org/2001/XMLSc	"Thompson"^^<http://www.w3.org/2001/XMLSchem	Thompson_Peter_I	CSC203
"Iniakpokeikiye"^^<http://www.w3.org/2001/XMLSc	"Thompson"^^<http://www.w3.org/2001/XMLSchem	Thompson_Peter_I	CSC311
"Iniakpokeikiye"^^<http://www.w3.org/2001/XMLSc	"Thompson"^^<http://www.w3.org/2001/XMLSchem	Thompson_Peter_I	CSC303
"Iniakpokeikiye"^^<http://www.w3.org/2001/XMLSc	"Thompson"^^<http://www.w3.org/2001/XMLSchem	Thompson_Peter_I	CSC301
"Iniakpokeikiye"^^<http://www.w3.org/2001/XMLSc	"Thompson"^^<http://www.w3.org/2001/XMLSchem	Thompson_Peter_I	CSC307
"Iniakpokeikiye"^^<http://www.w3.org/2001/XMLSc	"Thompson"^^<http://www.w3.org/2001/XMLSchem	Thompson_Peter_I	CSC305
"Iniakpokeikiye"^^<http://www.w3.org/2001/XMLSc	"Thompson"^^<http://www.w3.org/2001/XMLSchem	Thompson_Peter_I	CSC309

Figure 2. The following sparql query retrieve all the courses offered by a particular student.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ndu: <http://www.semanticweb.org/thompson/ontologies/2018/2/NDUStudent#>
SELECT ?StudentFirstName ?StudentLastName  ?MotherName
        WHERE {
                ?Student ndu:hasStudentBiodata ?BioData.
                ?BioData ndu:hasStudentParentInfo ?Mother.
                ?Mother ndu:mothersName ?MotherName.
                ?Student ndu:StudentFirstName ?StudentFirstName.
                ?Student ndu:StudentLastName ?StudentLastName.
              }
```

StudentFirstName	StudentLastName	MotherName
"Iniakpokeikiye"^^<http://www.w3.org/2001/XMLSchema#string>	"Thompson"^^<http://www.w3.org/2001/XMLSchema#string>	"Christopher, Toboulayefa"^^<http://www.w3.org/2001/XMLSchema#

Figure 3. Query that returns the 'mother' of a students including the students' first name and last name.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ndu: <http://www.semanticweb.org/thompson/ontologies/2018/2/NDUStudent#>
SELECT ?StudentFirstName ?StudentLastName  ?Hostel ?HostelRoom
        WHERE {
                ?Student ndu:hasStudentBiodata ?BioData.
                ?BioData ndu:hasAddress ?resAdd.
                ?resAdd ndu:HostelName ?Hostel.
                ?resAdd ndu:HostelRoomNo ?HostelRoom.
                ?Student ndu:StudentFirstName ?StudentFirstName.
                ?Student ndu:StudentLastName ?StudentLastName.

              }
```

StudentFirstName	StudentLastName	Hostel	HostelRoom
"Iniakpokeikiye"^^<http://www.w3.org/2001/XMLSc	"Thompson"^^<http://www.w3.org/2001/XMLSchen	"Hostel H"^^<http://www.w3.org/2001/XMLSchema	"9"^^<http://www.w3.org/2001/XMLSchema#int>

Figure 4. This query returns the Hostel and Hostel Room Number of students including the students' first name and last name.

6. EVALUATION

To determine the completeness of the developed ontology in relation to our purpose of translating the different databases in three the portals (university data management) and whether the ontology answers our intended sparql queries or not. We applied the tasked based approach to evaluate the ontology. Its performance is measured based on the way it performs in retrieving the relevant semantic information necessary for our purpose if the retrieved information is the desired semantic information. Hence, we semi-automatically populate the ontology by asserting some object properties into the ontology and execute some sparql queries. The query outputs were observed to be reasonably answering our intended purpose. Also, to ensure that the ontology is consistent, and its general qualities are meant at this level of development, we adopt the pellet and HermiT 1.3.8.418 reasoners to evaluate the ontology for consistency; and the ontology was observed to be consistent. However, at this initial evaluation, we did not evaluate the ontology for ontology-pitfalls (structural and lexical patterns) since the ontology development is still progress.

7. CONCLUSION

We have developed and implemented an adaptable ontology for university data management in the Niger Delta University. The target of this ontology is to prepare the ground for the advent of software agents and other applications that require more structured data for its computational processes as the institution grows to higher height. The resulting ontology captures the fundamental aspects of the selected university based relational databases from the different portals which are necessary for computational processes. The standard graph database query language Sparql was used in retrieving the necessary and specific information. However, our approach does not address certain salient challenging issues like result computation, words with different meanings, and implicit information not captured in the databases. In future our methodology will be automated in order to automatically map relational database schema elements as well as the database integrity constraints defined in the database to ontological classes, individuals and object properties.

REFERENCES

1. Antoniou, G. and Van Harmelen, F., 2004. *A semantic web primer*. MIT press.

2. Bak, J., Cybulka, J., and Jedrzejek, C. (2013) Ontological modeling of a class of linked economic crimes. Transactions on Computational Collective Intelligence IX, pp. 98–123. Springer, (2013)

3. Barbieri, D. F., Braga, D., Ceri, S., Valle, E. D., and Grossniklaus, M. (2010) C-SPARQL: a continuous query language for RDF data streams. *International Journal of Semantic Computing*, *4*(01), 3-25.

4. Berners-Lee, T., Hendler, J. and Lassila, O. (2001) The semantic web. *Scientific american*, *284*(5), pp.34-43.

5. Bezzazi, E. H. (2007) Building an Ontology That Helps Identify Criminal Law Articles That Apply to a Cybercrime Case. ICSOFT (PL/DPS/KE/MUSE) pages 179–185. 2007

6. Bizer, C., Heath, T., and Berners-Lee, T. (2011) Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts* (pp. 205-227). IGI global.

7. Deus, H. F., Veiga, D. F., Freire, P. R., Weinstein, J. N., Mills, G. B., and Almeida, J. S. (2010) Exposing the cancer genome atlas as a SPARQL endpoint. *Journal of biomedical informatics*, *43*(6), 998-1008.

8. Dimitrov, M. (2011) Semantic technologies and triplestores for business intelligence. In *European Business Intelligence Summer School* (pp. 139-155). Springer, Berlin, Heidelberg.

9. Fawei, B., Wyner, A., Pan, J.Z. and Kollingbaum, M. (2017) Using legal ontologies with rules for legal textual entailment. In *AI Approaches to the Complexity of Legal Systems* (pp. 317-324). Springer, Cham.

10. Fawei, B., Pan, J. Z., Kollingbaum, M. and Wyner, A. Z. (2018) A methodology for a criminal law and procedure ontology for legal question answering. In Joint International Semantic Technology Conference 2018 Nov 26 (pp. 198-214). Springer, Cham

11. Hassanzadeh, O. (2011) Introduction to Semantic Web Technologies & Linked Data. *University of Toronto*.

12. Hitzler, P., and Janowicz, K. (2013) Linked Data, Big Data, and the 4th Paradigm. *Semantic Web*, *4*(3), 233-235.

13. Horrocks, I. (2008) Ontologies and the semantic web. *Communications of the ACM*, *51*(12), pp.58-67.

14. Noy, N. F., and McGuinness, D. L. (2001) Ontology development 101: A guide to creating your first ontology.

15. Obitko, M. (2007) Translations between ontologies in multi-agent systems. *Unpublished Ph. D. dissertation, Faculty of Electrical Engineering, Czech Technical University in Prague*.

16. Osathitporn, P., Soonthornphisaj, N., and Vatanawood, W. (2017) A scheme of criminal law knowledge acquisition using ontology. Software Engineering, Artificial Intelligence,

Networking and Parallel/Distributed Computing (SNPD), 2017 18th IEEE/ACIS International Conference on, pp. 29–34. IEEE (2017).

17. Pérez, J., Arenas, M., and Gutierrez, C. (2006) Semantics and Complexity of SPARQL. In *International semantic web conference* (pp. 30-43). Springer, Berlin, Heidelberg.

18. Ristoski, P. and Paulheim, H. (2016) Semantic Web in data mining and knowledge discovery: A comprehensive survey. *Journal of Web Semantics*, *36*, pp.1-22.

19. Saleem, M., Khan, Y., Hasnain, A., Ermilov, I., and Ngonga Ngomo, A. C. (2016). A fine-grained evaluation of SPARQL endpoint federation systems. *Semantic Web*, *7*(5), 493-518.

20. Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., Katz, Y. (2007) Pellet: A practical owldl reasoner. Web Semantics: science, services and agents on the World Wide Web, 5(2): pages 51–53. 2007 Elsevier.

21. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32nd annual meeting on Association for Computational Linguistics, pp. 133–138 (1994)

22. Wyner, A. (2008) An ontology in OWL for legal case-based reasoning. Artificial Intelligence and Law, 16(4) pages 361. 2008 Springer.