

Ontology Languages, Development Tools and Repositories: Towards a Unifying Platform

¹A. A. Obiniyi, ²O. N. Oyelade, ³G. I. Auta

^{1,2}Department of Mathematics, Ahmadu Bello University Zaria, Kaduna State, Nigeria

³Waziri Umaru Federal Polytechnic, Birnin-Kebbi, Kebbi, Nigeria

¹aaobiniyi@gmail.com; ²olaide_oyelade@yahoo.com; ³ockanya@gmail.com

Abstract-Knowledge representation enabling inference making has been greatly promoted through the Semantic Web. The Semantic Web provides and supports some ontology languages that have been used in representing machine readable knowledge base. Furthermore, both classical and newer ontology development tools abounds correspondingly with the available ontology languages. Also, storage models and systems have been proposed and used in storing ontology files. This research first reviews some ontology languages and their development tools, and then discusses some ontology repositories. Finally, a proposed model for unifying the task of ontology creation and editor, and as well as incorporating this development tool with ontology repository is presented. We indicate that storing such ontology in the proposed repository does make it lose its semantics.

Keywords- DARPA; RDF/RDFS; DAML; OIL; OWL; XTM; OWLIM; Sesame

I. BACKGROUND OF THE STUDY

Ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary [1]. It could be seen as a means of capturing concepts, entities, things that are peculiar to a specific domain of interest and then forming a relation among these concepts or things in such a way that link of data (dataset) would have been created. This dataset encapsulates all necessary information relevant to that domain. In general, ontology provides mechanism to capture information about objects or classes and the relationships that hold between them in a domain of interest. Language is a platform upon which generalization of acceptable means of communication is been realized. Hence several ontology languages have been designed and used in modeling ontology for any domain of interest. And some of such ontology languages include: Resource Description Framework/Resource Description Framework Schema (RDF/RDFS), Defense Advanced Research Projects Agency (DARPA), DARPA Markup Language + Ontology Inference Layer (DAML+OIL), Ontology Web Language (OWL), eXtensible Markup Language (XML), XML Topic Map (XTM) and Web Service Modeling Ontology language (WSMO).

Primitive ontology languages are Simple HTML Ontology Extension (SHOE), Ontobroker and Ontology Inference Layer (OIL) [2]. Several ontology languages have evolved and coordinated by different organizations, such W3C and US Department of Defense. Afterward, some other ontology languages that were designed in pursuit of enhancing and incorporating reasoning to ontology are RDF/RDFS, DAML, OIL, OWL, XTM and WSMO. RDF is a domain independent ontology language the makes no assumptions about a particular domain of use, and the user is required to define the needed terminology in a schema language called RDFS [3]. Hence RDF/RDFS are jointly used to define concepts of a given domain. DARPA sponsored DAML-ONT and OIL were later merged to form what was then referred to as DAML+OIL and were later submitted to World Wide Web (W3C) [4]. OWL is another ontology language that was designed and been coordinated by W3C. OWL was a bid to improve on the expressiveness of RDF/RDFS by the W3C. OWL has three variants which are Web Ontology Language Full (OWL) Full, Web Ontology Language Lite (OWL Lite) and Web Ontology Language Defeasible Logic (OWL DL). OWL2 has OWL2-RL, OWL2-QL and OWL2-EL as the three variants it consists of. Another ontology language is XMT and was originally designed to handle the construction of indexes, glossaries, thesauri and table of contents. Lastly, WSMO is an ontology used for modelling services that are been searched [5].

A major characteristic of data repository is its provision for the manipulation and query of the underlying data. Quite a number of semantic web query languages have emerged are tailored to either a particular ontology language or related ontology languages. And some of these query languages include Sparql Protocol and RDF Query Language (SPARQL), Semantic Query Web Rule Language (SQWRL) and RDF Query Language (RQL). SPARQL queries are pattern matching queries on triples that constitute an RDF data graph [2]. SQWRL is a query language used in querying OWL and are built interoperability with a rule language called Semantic Web Rule language (SWRL) in mind. They take the antecedent of SWRL and effectively treat it as a pattern specification for a query [6] RDF Query Language (RQL) is a query language used to query RDF. It illustrates a number of features that will be part of any reasonable query language in RDF and RDFS, such as path expressions and schema awareness [4].

II. A COMPARATIVE STUDY OF ONTOLOGY LANGUAGE

XML models its data in a hierarchical pattern. Data Description Document (DTD) and XML Schema are used in validating languages or formats of XML files. It is a subset of SGML and its goal is to enable generic Standard Generalized Markup Language (SGML) to be served. Several query languages have been designed for extracting data from XML and one of such is XPath [4]. RDF on the other hand, is used to model information in the form of a 'graph'. Basically, it represents data in a triple format, as in subject-predicate-value. RDF evolved as a result of lack of semantics in XML, and it is one of the pillars of Linked Data Web or the Semantic Web. XML Topic Maps (XTM) is another ontology language that stems from XML. Though XTM is an evolving ontology language, it has not received much public attention as OWL [7].

OWL, also referred to OWL 1, enables users to develop explicit and, formal conceptualizations of domain models. OWL Lite, OWL DL, and OWL Full are three sub-languages of OWL. OWL Lite is the simplest of all the sub-languages and is appropriate for use when simple class hierarchy and constraints are needed. OWL DL is more expressive than OWL Lite and is based on description Logics. Computing the classification hierarchy and check for inconsistencies in an ontology that conforms to OWL DL becomes easier because description logic is a decidable fragment of first order logic and is amenable to automated reasoning [8]. OWL Full is the most expressive of all the three sub-languages of OWL. It places higher priority on high expressivity of concepts so as to guaranty the computational completeness of the language.

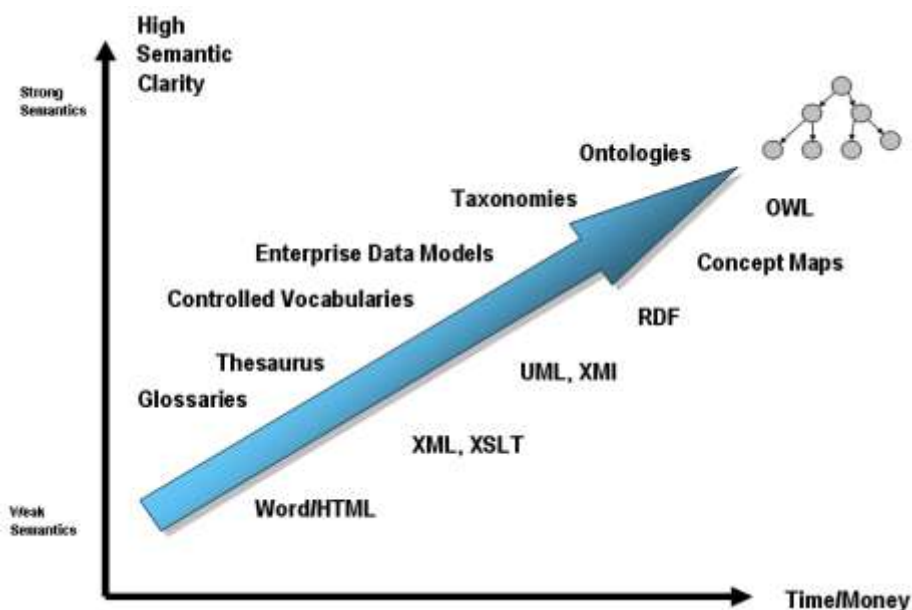


Fig.1 A graphical illustration of some markup and ontology languages [9]

OWL 2, an ontology language, is designed to make OWL 1 more improved so as to make ontology development richer. OWL 2 ontologies are sometimes referred to as RDF graph, and can be employed for use with RDF graph. Virtually all OWL 2 tools support its exchange in RDF/XML format and its corresponding serialization formats. Semantically, OWL 2 has two different ways of reading meaning into its ontologies. These are the Direct Semantics and RDF-Based Semantics, and these two different semantics are utilized by semantic web tools and reasoners' requests or queries. While the Direct Semantics provides direct meaning into ontology structure, the RDF-Based Semantics provides reading meaning into the RDF graph. However, there exists a link between these two semantics of OWL 2 which states thus: inferences generated from an OWL 2 DL using Direct Semantics will still be correct if the resulting ontology is mapped into an RDF graph and the graph is read using the RDF Semantics. Interestingly, OWL 2 also has three sublanguages or profiles just like its preceding OWL 1 language. OWL 2 RL, OWL 2 QL and OWL 2 EL are the profiles of OWL 2. Technically, these profiles are designed to compromise high expressivity for efficient reasoning tasks which include querying, subsumption and class consistency. OWL 2 QL are best used for applications whose underlying ontology is made up of large number of instances that will be queried, and this reduces expressivity of the profile. In addition, OWL 2 QL also allows information stored on a relational database to be queried alongside querying its own ontology. OWL 2 RL is mostly used in conjunction with rule language, and this enhances expressivity for RDF based applications. Its reasoning tasks are implemented using some rule engines. The last OWL 2 sublanguage is OWL EL. OWL EL profile provides ontology developers with the logic that provides existential quantification. It is used for applications that basically need high number of properties and classes, and it provides sufficient expressive power for ontologies that use them [10]. Generally speaking, OWL 2 shares some similarities with OWL 1. However, it provides some constructions that both improve expressivity (richer data

types, property chains, keys, qualified cardinality restrictions and enhanced annotation capabilities) and syntactic sugar (as in disjoint union of classes).

To conclude this section, we note that OWL, particularly OWL 2 is more efficient and powerful in ontology development and even in wide use compared to other ontology languages discussed in this section. However, even in the choice of OWL, the user must technically make a choice of what sublanguage to use given the type of semantic web application that will be used with it.

RDF/RDFS have support for rules and this is supported by a Java-based framework called Jena API [11]. OWL has support for rule language like Semantic Web Rule Language (SWRL) [12]. Semantic Query Web-enhanced Rule Language (SQWRL) is a query language that is primitive to OWL and usually results in high search hits compared to SPARQL. SPARQL has been used in querying RDF and OWL [13], however, SPARQL's understanding of OWL construct is incomplete [6] and this may affect output of query when SPARQL is used on it.

A general study of available ontology languages and their developing tools is necessary in the choice of an appropriate ontology language to use in a specific domain of choice. It has been observed that some of these ontology languages are more efficient in some domain compared to other domain. Furthermore, the choice of an ontology language will likewise affect the choice of the rule language and query language that will be used. Hence, a study and choice of an ontology language is necessary to consider the supporting technologies that will work in handy of that ontology language.

III. MODELING TOOLS, ONTOLOGY EDITORS AND SEMANTIC WEB REPOSITORIES

Manual development of ontology files can be error prone and with more inconsistencies. In this section, some ontology editors and modelling tools are considered with a view to highlight their viability and other supports they provide ontology developer with during the process of ontology development.

Ontolingual Server: is the first ontology development tool [14]. This ontology development tool was actually meant for ontolingual IDL KIF ontology language. IDL KIF is an ontology language with minimal scope of usage.

Ominigator and Ontopia: is an ontology development tool designed for XTM. Ontopoly is built as a client/server application. From the client side, developers are provided with use the web browser while the server is a web server bundled with the distribution. The server-side application is built using the Navigator Framework and Web Editor Framework, which are part of the Ontopia knowledge Suite. Ontopoly is accessible through Ontopia- an application that provides easy access to Ontopoly, Ominigator and Vizigator.

OntoEdit: This is an ontology engineering environment that was developed by Institute of Applied Informatics and Formal Description Methods (IAIFDM) at the Karlsruhe University. The ontology can be exported to several ontology languages such as OntoEdit Extensible Markup Language (OXML), Flogic, RDF(S), and DAML+OIL. It supports rules and other plugins. However, it is not an open source.

OilEd: this is an ontology editor that has support for OIL. The design is targeted towards demonstrating how it can be extended to deal with a more expressive modeling language and how reasoning can be used to support ontology development.

WebODE and WebOnto: Web Ontology Development Environment (WebODE) is three-tier architecture and provides platform for XML, RDF(S) and CARIN alongside the capability to export ontology into languages such as XML, RDF(S), CARIN, and DAML+OIL. WebOnto is client/server architecture. It is used for developing ontology languages like Operational Conceptual Modeling Language (OCML), RDF(S) and OIL.

Protégé is an open source ontology development tool. Protégé was developed by Stanford Center for Biomedical Informatics Research at the Stanford University (<http://protege.stanford.edu/>). This ontology development tool is used for developing OWL, RDF and XML. Protégé is a powerful tool that permits other third party software to be ported or plugged into it so as to enhance the ontology that has been developed. Some of the plug-in allowed in protégé are Jess, FaCT, and HERMIT. Some of these plug-in provide querying, reasoning, rule implementation, visualization and some other functions. Protégé provide graphical user interface that users can keep clicking in the process of creating ontology. WebProtégé- is the online version of Protégé [15].

Other ontology editors includes: SWOOP, a simple and small ontology editor, Vitro, a semantic web application with an incorporated ontology editor. Top Braid Composer provides a semantic web editor that gives room for several ontology languages to be created and edited. Pool Party (Vocabulary modeler) is a SKOS editor with a thesaurus management system. Neon Toolkit is an ontology editor that is fitting for ontology consisting of very large concepts. And like Protégé it allows many plugins. Altova Semantic Works, provides users with a platform to create XML ontology files and as well as access them for transformation and editing purposes. Visual Ontology Modeler, is a UML-based visual ontology developer and it enables users to design their desired ontology on component-based approach. Neologism is a vocabulary editor that is online and at the same time helps ontology

developers to publish their work. Ontofly also is web-based editor like Neologism. A good knowledge-based editor with community driven usage is Knoodl. Hozo Ontology Editor.

A good choice of ontology developing software and editors helps developers in designing a more correct ontology for applications. Several ontology editors in cooperates other plugins that onerously help developer improve their ontology design, visualize it, and check for consistency of information modelled. It can be used to develop both simple and complex ontology-based applications. And since they also help in developing intelligent systems, they allow rule systems – a combination of rule engine and rules implemented with semantic web rule languages – so as to yield a wide range of intelligent systems. Protégé provides supports such as DIG reasoned, Pellet, check for consistency, check for taxonomy, computation of inferred types and ontology visualizer plug-in like OWLVIZ. It is open source software, with large online support groups who develop ontology for applications in e-commerce, biomedicine, organizational and institutional purposes. Protégé fully supports OWL 1 and the OWL 2 discussed in this paper, and RDF. Whereas other ontology editors have some limitations that disadvantage them in enjoying wide user community, Protégé continues to be in use by many developers. For instance, Ominigator is strictly used for creating XTM ontology, OntEdit, though still in use, but not as pronounced as Protégé. Furthermore, we noted that OWL is one of the most powerful and popular ontology language, WebODE however, does not have provision for creating OWL ontologies in it, though it supports RDFS and some other ontology languages such as OCML. Whenever a developer is in search of ontology development software or editor, the likes of Protégé might be their best bet.

On the other hand, ontology repository serves as a medium for store and search of semantic web based modeled facts. OWLIM, 4Store, Bigdata and Sesame are examples of ontology repositories. OWLIM, now referred to as GraphDB, is the most pronounced of them all because of its scalability. It has two editions which are SwiftOWLIM and BigOWLIM. Some of the features provided by OWLIM include inference engine, triple store and SPARQL query engine. It is also seen as RDF database management systems with strong support for the semantics of OWL Horst, OWL 2 RL, RDFS and OWL 2 QL. And it uses TRREE engine in handling OWL 2 RL, OWL 2 QL, OWL Horst reasoning, and RDFS. Because of its biasness towards RDFS, this pegs OWL 2 RL as the most expressive language, it supports since it (OWL 2 RL) has RDFS [16]. OWLIM provides support for optimizing query, and also helps in consistency checking. It is packaged as a storage and inference layer (SAIL) for Sesame. OWLIM (GraphDB) has three versions: GraphDB Lite, GraphDB Standard and GraphDB Enterprise. Basically, GraphDB Standard has more provision for the storage of RDF statements compared to GraphDB Lite. While GraphDB Standard allows billions of RDF statements to be stored, GraphDB Lite only permits users to load a maximum of hundred million statements. GraphDB Lite is the fastest repository with reliable data integrity, and allows loading and querying of triple stores to be done in the memory rather than the file based operation. Meanwhile, GraphDB Standard incorporates powerful inference engines for enhanced reasoning. It also includes the functionality for checking the sameness of different concepts across different data sources. GraphDB Enterprise, however, can run on multiple servers, coupled with the features of GraphDB Standard it is made of. It offers parallel scalability in query performance across servers. GraphDB (OWLIM) offers users with such enormous loading of data sources and as well the execution of query and inference making on those data sources. However, Sesame is an open source RDF framework implemented with Java. It provides SPARQL endpoints and triple stores through its built-in server. This SPARQL endpoint enables the query (and even federated query) and update of SPARQL, specifically, SPARQL 1.1, with its inferencing and querying support biased towards RDFS. And it offers a considerable number of tools for exploring the usefulness of RDF/RDFS. As stated above, it provides SAIL API for other stores (third parties, OWLIM is an example) to use it. When considering the choice of ontology store, OWLIM will seem attractive because of its wide range of supports and features it offers to its users.

Meanwhile, this paper seeks to promote a unified platform for the tasks of ontology development, editing, deployment (query, applications usage, and inferencing) and storage. Consequently, some ontology development systems (e.g. Protégé) discussed above is already coupled with features/tools that provide ontology developer developers a means to design semantic web related applications. By the way, we acknowledge the inference making, rule supports, visualizing tabs and other ontology developmental tools provided as plug-ins to such systems. However, we canvass for more systems that harness the multiple tasks of developing, editing, querying, inferencing, storing and the use of ontologies. Hence, we propose a model that illustrates our unified platform.

IV. A UNIFIED MODEL FOR ONTOLOGY DEVELOPMENT AND STORAGE

Considering the fact that much have been done in the areas of ontology language editors/development tools, and as well with regard to ontology repository, we do not seek to suggest new ones but to propose a systematic harmonization of these two major tools and their a supportive features, for the purpose of providing ontology developer with multi-purpose platform that explores and interactively passes ontologies through most of its compartments, with the sole aim of improving ontology development and maintenance. Our proposed model includes two major modules: ontology editor and storage. Each of these modules has its containing compartments with which they employ to aid their productivity. Fig. 2 captures each of these compartments.

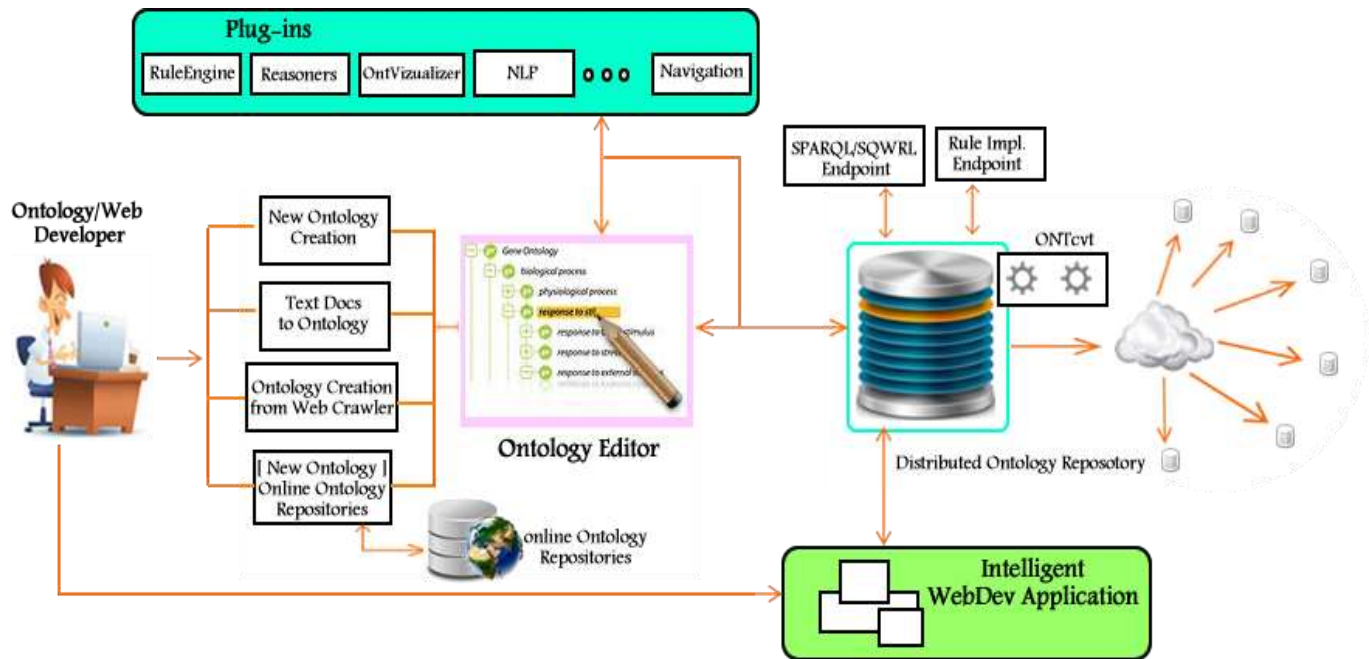


Fig. 2 Unified Ontology Creation and Management Platform

The first module consists of four other components. These include the new ontology creation, text-docs-to -ontology, ontology creation from web crawler and creation of ontology from existing online repositories of ontologies. The last one involves ontology developer adopting a similar ontology having the same domain with the one the developer intends to develop, except that such ontology can be edited to need of required user. The third component allows developers to task web crawlers to crawl over Linked Data and generate a semantic web graph or ontology files that will subsequently be edited to taste. The ontology developer will invoke the crawler through a web search. There exists mechanism for intelligently mining text from a document. Relevant information mined can then be strategically controlled by the ontology developer in creating or developing desired ontology from it, and this is what the second component does. Lastly, the first component is the most likely functionality that virtually all ontology editors discussed in Section 3.1 provide their users with – creation of ontology scratch. Observe that each of the tasks of these components can be accessed through the ontology editor.

Ontology repository module is captured on the right hand side of the model in Fig. 2. We assume distributed nodes for the repository server. This is to give room for scalability and manage node failure through the load balancing mechanism. Depending on the type of ontology language (specifically RDF/RDFS and OWL) been used on the ontology editor side, the ONTcvt component ensures that the semantics of the ontology is not lost during storage and retrieval. Query end points are provided for SPARQL and SQWRL queries. More so, rule implementation access point is made available through the Rule Imp end point in Fig. 2. The array of plug-ins provides their varying and respective supports to the two modules. Also, the semantic web application development platform is shown in Fig. 2 below. Web applications developed from this medium could be deployed with a specific ontology(s) from the repository as their data source.

Finally, we reason that an encompassing semantic web platform for the creation and management of ontologies does not portend any unnecessary congestion of functionality, but to provide developers with all-tools system in their pursuit the demanding task of ontology development. This model can be made available to developers when some of these components (which already are in existence) are intelligently coupled together.

V. CONCLUSION

Ontology languages are good data modelling and efficient knowledge representation languages. They have helped in modelling entities of any domain of choice. Intelligent agents and systems have been built to read and understand information model in ontologies. These intelligent agents are able to reason over such knowledge base. It must also be noted that the larger the ontology to be developed, the more inconsistent it tends to be when created manually. Hence, in this paper, we reviewed some ontology development tools and editors. Furthermore, considered some available ontology repositories as a way of enlightening readers about what is available. And finally, we propose a unifying model for all these tools.

REFERENCES

- [1] Neches R., Fikes R. E., Finin T., Gruber T. R., Senator T., and Swartout W. R., "Enabling technology for knowledge sharing," *AI Magazine*, vol. 12, no. 3, pp. 36–56, 1991.
- [2] Neumann T. and Gerhard W., "RDF3X: a RISC-style Engine for RDF," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 647-659, 2008.
- [3] Antoniou G. and Frank V. H., *Semantic Web Primer*, MIT Press, Cambridge, Massachusetts, London, England, pp. 113-114, 2004.
- [4] Fensel D., Horrocks I., van Harmelen F., Decker S., Erdmann M., and Klein M., Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling, and Management (EKAW 2000). OIL in a Nutshell. In: *Lecture Notes in Artificial Intelligence*, pp 1–16, 2000.
- [5] Sbodio M. L., David M., and Claude M., "Discovering Semantic Web Services Using SPARQL and Intelligent Agents," *Elsevier Journal*, p. 311, 2010.
- [6] O'Connor M., and Amar D, SQWRL: a Query Language for OWL. In *OWL: Experiences and Directions (OWLED)*, 5th International Workshop pp. 1-8.
- [7] Park J. E., and Sam T. E. H., *XML Topic Maps: Creating and Using Topic Maps for Web*, Addison Wesley, 2002.
- [8] Horridge M., S. Jupp, G. Moulton, A. Rector, R. Stevens, and C. Wroe, *A practical guide to building OWL ontologies using the Protege 4 and CO-ODE tools (Edition 1.1.)*, University of Manchester, 2007.
- [9] Dimitrov M., Third (3rd) Gate Training Course, Ontotext, RDF, SPARQL, and Semantic Repositories, October 9, 2014, retrieved from http://www.slideshare.net/marin_dimitrov/rdf-sparql-and-semantic-repositories.
- [10] OWL 2 Web Ontology Language, retrieved on September 16, 2014, from <http://www.w3.org/TR/2012/REC-owl2-overview-20121211/>
- [11] Apache Jena, retrieved May 26, 2013, from <http://www.apache.org/index.html>.
- [12] O'Connor M., and Amar K. D., "A Lightweight model for representing and reasoning with temporal information in biomedical ontologies," *HEALTHINF- Proceedings of the Third International Conference on Health Informatics*, pp. 90-97, Valencia, Spain.
- [13] SPARQL, retrieved October 9, 2013, from <http://www.w3.org/TR/rdf-sparql-protocol/>
- [14] Farquhar A., Fikes R., and Rice J., "The Ontolingua Server: A Tool for Collaborative Ontology Construction," *Intl. J. Human-Computer Studies*, vol. 46, pp. 707–728, 1977.
- [15] Protégé retrieved October 4, 2013, from <http://protege.stanford.edu/>
- [16] OWL 2 Web Ontology Language Profiles Retrieved on September 16, 2014, from <http://www.w3.org/TR/2012/REC-owl2-profiles-20121211/>

Obiniyi A. A received his Ph. D degree in Computer Science from Ahmadu Bello University, Zaria in Kaduna, State of Nigeria in 2009. He is a member of Nigeria Computer Society, Internet Society and a Chartered member of Computer Professionals (Registration Council of Nigeria). He lectures in the Department of Mathematics of ABU and has many publications to his credit.

Oyelade N. Olaide is a graduate student of computer science at the Department of Mathematics Ahmadu Bello University, Zaria-Nigeria. His current research interest includes Semantic Web technologies, specifically ontology languages and their query and rule languages. Also, he has research interest in mobile computing. Oyelade earned his B.Sc. degree in computer science from University of Jos, Nigeria in 2010.

Autu. I. Gambo obtained his first degree in Computer Science from Usman Danfodiyo University Sokoto, Nigeria. And he is a graduate student of Computer Science at the Department of Mathematics, Ahmadu Bello University, Zaria, Nigeria. He is presently lecturing at Waziri Umaru Federal Polytechnic Birnin Kebbi.