Digital Innovations & Contemporary Research in SCIENCE, ENGINEERING & TECHNOLOGY

Vol. 6. No. 2, June, 2018

Article Citation Format
Sunday, Kissinger, R.G. Jimoh, Agbo, Friday & Babandi Fatima Shuaibu (2018). A Proposed Paradigm For E-Commerce Based On Semantic Web Technologies Journal of Digital Innovations & Contemp Res. In Sc., Eng & Tech. Vol. 6, No. 2. Pp 1-14

Article Progress Time Stamps

Article Type: Research Article
Manuscript Received: 15[rd] November, 2017
Review Type: Blind
Final Acceptance:: 5[th] December, 2017
DOI Prefix: 10.22624

A Proposed Paradigm For E-Commerce Based On Semantic Web Technologies

Sunday, Kissinger, R.G. Jimoh (Phd), Agbo, Friday & Babandi Fatima Shuaibu
[1]Department of Mathematics, Computer Science (Unit), Usmanu Danfodiyo University, Sokoto, Nigeria
[2&3]Dept. of Computer Science, University of Ilorin, Ilorin, Nigeria
[4]Dept. of Computer Science, College of Education, Jigawa State.
E-mails: sundaykissy@gmail.com; jimoh_rasheed@unilorin.edu.ng; freddy4uagain@gmail.com; Fatima.babandi@gmail.com

ABSTRACT

Several web applications have been deployed by business enterprises through which their products would not only be made available on the internet, but also enable their prospective consumers to be able to follow some procedures to make their purchases online. This is normally achieved with the help of semantic web technologies, namely Resource Description Framework (RDF), Resource Description Framework Schema (RDFS), DARPA Agent Markup Language (DAML) plus Ontology Inference Layer (OIL) and Web Ontology Language (OWL 1). This paper takes advantage on some of the Semantic Web technologies mentioned in the previous sentence in order to design a model for enhancing Business to Consumer (B2C) e-commerce applications. In this paper the researchers developed OWL 2 ontology rich in more expressive OWL constructs for rich entailments. Qualified Cardinality Restriction (QCR) which OWL 2 is known for is also applied to the ontology. Also in this paper the researchers compared two popular reasoners in querying the underlying ontology in a programmatic way. Various forms of ontology display was shown to gain more understanding of our ontology. The overall focus of this research is to provide a more efficient framework for B2C through the right application of Semantic Web Technologies.

Key words: Ontology, Semantic Web, E-commerce, RDF, OWL.2

1. BACKGROUND TO THE STUDY

Business-to-consumer (B2C) electronic commerce is the predominant commercial experience of Web users. It is changing the way businesses interact with consumers, as well as the way they interact with each other. A typical scenario involves a user visiting one or several online shops, browsing their offers, selecting and ordering products (Antoniou & Harmelen, 2004). Electronic interactions are increasing the efficiency of purchasing, and are allowing increased reach across a global market. The inability of software agents to carry out tasks on behalf of the user poses serious setback to purchasing commodities from online stores. Ideally, a user would collect information about prices, terms, and conditions (such as availability) of all, or at least all major online shops and then proceed to select the best offer. But manual browsing is too time-consuming to be conducted on this scale. Typically a user will visit one or a very few online stores before making a decision.

To alleviate this situation, tools for shopping around on the Web are available in the form of shopbots, software agents that visit several shops, extract product and price information, and compile a market overview. Their functionality is provided by wrappers, programs that extract information from an online store. One wrapper per store must be developed. This approach suffers from several drawbacks: The information is extracted from the online store site through keyword search and other means of textual analysis. This process makes use of assumptions about the proximity of certain pieces of information (for example, the price is indicated by the word price followed by the currency symbol then followed by a positive number). This heuristic approach is error-prone; it is not always guaranteed to work. Because of these difficulties only limited information is extracted. For example, shipping expenses, delivery times, restrictions on the destination country, level of security, and privacy policies are typically not extracted. But all these factors may be significant for the user's decision making. In addition, programming wrappers is time-consuming, and changes in the online store outfit require costly reprogramming.

The main obstacle to providing better support to Web users is that, at present, the meaning of Web content is **not machine-accessible**. Of course, there are tools that can retrieve texts, split them into parts, check the spelling, and count their words. But when it comes to interpreting sentences and extracting useful information for users, the capabilities of current software are still very limited. An alternative approach is to represent Web content in a form that is more easily machine-processable and to use intelligent techniques to take advantage of these representations. We refer to this plan of revolutionizing the Web as the **Semantic Web initiative** (Antoniou & Harmelen, 2004). The Semantic Web is propagated by the World Wide Web Consortium (W3C), an international standardization body for the Web (Berners-Lee T. , 1994). The driving force of the Semantic Web initiative is Tim Berners-Lee, the very person who invented the WWW in the late 1980s.

The development of the Semantic Web has a lot of industry momentum, and governments are investing heavily. The U.S. government has established the DARPA Agent Markup Language (DAML) Project, and the Semantic Web is among the key action lines of the European Union's Sixth Framework Programme. Other semantic technologies are the Resource Description Framework (RDF), Resource Description Framework Schema (RDFS) Web Ontology Language version one (OWL 1) and Web Ontology Language Version 2 (OWL 2).

1.1 Statement Of Problem

Electronic commerce is having a revolutionary effect on business. It is changing the way businesses interact with consumers, as well as the way they interact with each other. Electronic interactions are increasing the efficiency of purchasing, and are allowing increased reach across a global market. According to our knowledge, current literatures on harnessing the power of semantic web technologies in developing business to consumer e-commerce applications still rely on the primitive aspect of these technologies such as the Resource Description Framework (RDF), Resource Description Framework Schema (RDFS), DARPA Agent Markup Language (DAML) plus Ontology Inference Layer (OIL) in creating and developing our ontology for B2C e-commerce (Oyelade, Junaidu, & Obiniyi, 2014). Just of recent it was discovered that even the "almighty" Web Ontology Language (OWL 1) possesses expressivity limitations, qualified cardinality constraint, syntax issues, and other problems (Grau, et al., 2008). In order to develop a more vibrant and expressive ontology that will facilitate ontology development and sharing via the web, with the ultimate goal of making web content more accessible to machines; the need to employ OWL 2 becomes a necessity.

2. LITERATURE REVIEW

Semantic, from the Greek word "semantikos", involves giving significance or meaning to words or symbols, thus enabling distinctions to be made between the meanings of different words or symbols **(Valencia, Rodríguez, & Ricardo, 2014)**.

The semantic web **(Berners-Lee et al., 2001)** was conceived with the statement "I have a dream for the Web". It aims at adding semantics to the data published on the Web (i.e. establish the meaning of the data), so that machines are able to process these data in a similar way humans do. For this, ontologies are the backbone technology **(Sanchez, Valencia-Garcıa, Bejar, & Breis, 2009).** The semantic web is sometimes used to indicate the varicose technologies that it supports. The diagram in Figure 1 illustrates how these technologies relate with each other. This is well known as the Semantic Web cake.

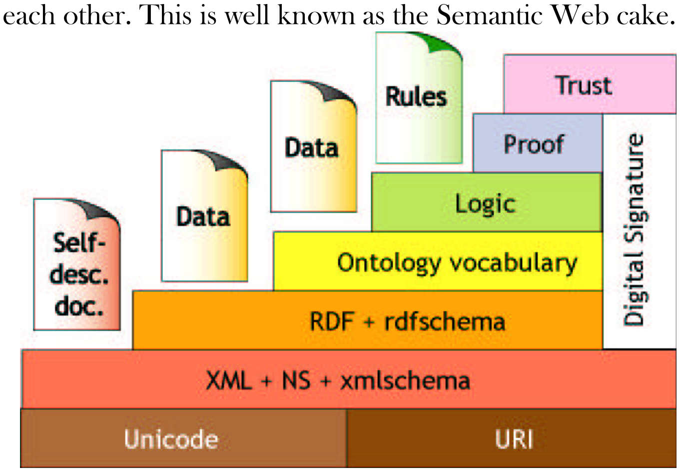

Figure 1. Semantic Web Technology Stack. Ian and Peter (2003)

According to **(Antoniou & Harmelen, 2004)**, semantic web originates from philosophy. In that context, it is used as the name of a subfield of philosophy, namely, the study of the nature of existence, the branch of metaphysics concerned with identifying, in the most general terms, the kinds of things that actually exist, and how to describe them. However, in more recent years, ontology has become one of the many words hijacked by computer science and given a specific technical meaning that is rather different from the original one. According to **T. R. Gruber's** definition which was later refined by **R. Studer:** An ontology is an explicit and formal specification of a conceptualization.

In general, an ontology describes formally a domain of discourse. Typically, an ontology consists of a finite list of terms and the relationships between these terms. The terms denote important concepts (classes of objects) of the domain. Since the inception of the Semantic Web, the development of languages for modeling ontologies— conceptualizations of a domain shared by a community of users—has been seen as a key task. The initial proposals focused on RDF and RDF Schema; however, these languages were soon found to be too limited in expressive power (I., Patel-Schneider, & van Harmelen, 2003). Moreover, such descriptions should be amenable to automated reasoning if they are to be used effectively by automated processes, e.g., to determine the semantic relationship between syntactically different terms. The recognition of these requirements led to the development of DAML+OIL, an expressive Web ontology language. DAML+OIL is the result of a merger between DAML-ONT, a language developed as part of the US DARPA Agent Markup Language (DAML) programme and OIL (the Ontology Inference Layer)

(Fensel, Harmelen, Horrocks, McGuinness, & Patel-Schneider., 2001), developed by a group of (mostly) European researchers.
Subsequently the World Wide Web Consortium (W3C) went ahead to form the Web Ontology Working Group, whose goal was to develop an expressive language suitable for application in the Semantic Web. The result of this endeavor was the OWL1 Web Ontology Language, which became a W3C recommendation in February 2004. OWL1 is actually a family of three language variants (often called species) of increasing expressive power: OWL Lite, OWL DL, and OWL Full (Patel-Schneider, Hayes, & Horrocks, 2004).

Despite the success story surrounding OWL1, the numerous contexts in which the language has been applied have revealed some deficiencies in the original design (Grau, et al., 2008). For example, ontology engineers developing ontologies for biomedical applications have identified significant expressivity limitations of the language. Also, the designers of OWL APIs have identified several practical limitations such as difficulties in parsing OWL ontologies or the inability to check for obvious errors, such as mistyped names (Grau, et al., 2008). As a response, the community of OWL 1 users and application designers developed various patterns for approximating the missing constructs. Since the actual expressive power is missing, these workarounds are often unsound or incomplete with respect to the intended semantics (Grau, et al., 2008).

3. METHODOLOGY

3.1 Introduction

In this work, two ontologies were developed. The first ontology which is the core ontology, serves as a registry for retailers to make their ontologies known to intending searchers. The core ontology however, provides information about retailers and most importantly it provides a link to the ontology of the retailers so that users requests may be searched against the retailers ontologies. The second ontology is the retailer ontology, where information such as the price, model, and quantity of an available product will be made available by every individual retailer for their prospective consumers. The researchers however, constrained the research to model the Laptop domain.

3.2 Architectural Framework of the System

Figure 2.0 represents the architecture of the system. The architecture however comprises the following components divided into 3 layers:-

- **Server Layer:** This layer contains the following components;
- **Core Ontology**: This component contains the names, addresses, contact numbers, URLs of retailers that have registered with the server and they are implemented in an OWL file. Logic Tier: All Java classes written to translate the user request into SQWRL, control the flow of the agent on the server side and for inference making with HermiT reasoner and OWL API.
- **Reasoning**: The HermiT inference engine and Jess rule engine are used here.
- **OWL API**: This API designed for handling OWL2 ontologies is used in order to enhance our Java codes in accessing the OWL2 file.
- **Retailer Layer:** This layer is composed of the retailer ontology which contains the products in the retailer's store.
- **User (Consumer) Layer:** This layer has just one component.
- Registration, Login and Request Forms: These forms are used for registration, login and entering user requests.

- Logic tier: This tier provides the basic Java classes that drive the functionalities provided by the mobile device.

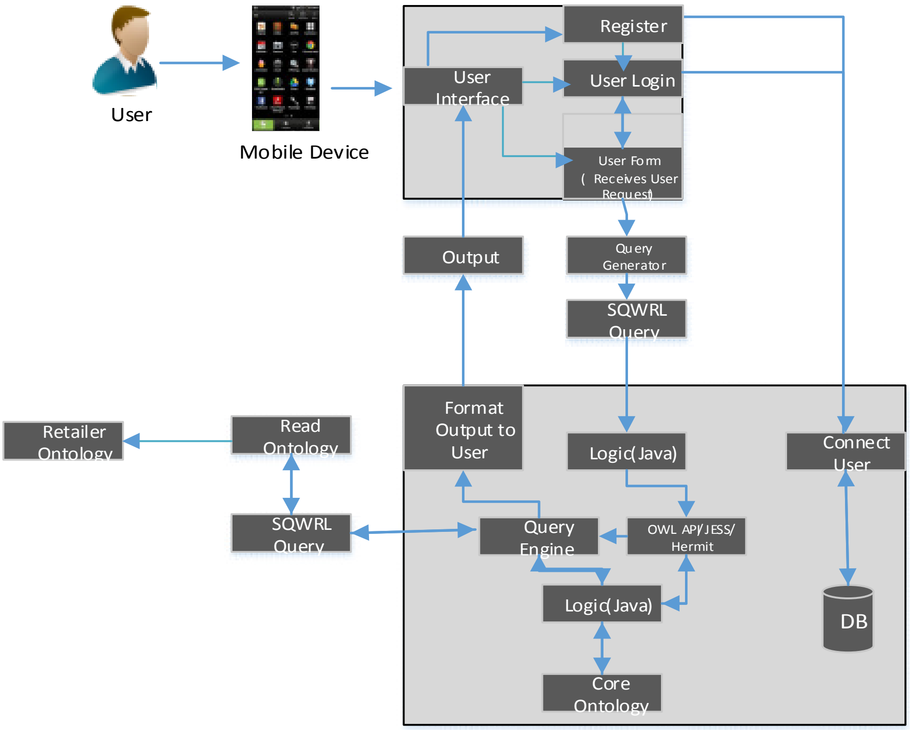

Fig 2.0: System Architecture

3.3 Ontology Design

OWL 2 has more expressivity than OWL 1. OWL 1 provides very limited expressive power for describing classes whose instances are related to concrete values such as integers and strings. In OWL 1, it is possible to express restrictions on datatype properties qualified by a unary datatype. For example, one could state that every British citizen must have a passport number which is an xsd:string, where the latter is an XML Schema datatype—a unary predicate interpreted as the set of all string values. In OWL 1, however, it is not possible to represent restrictions to a subset of datatype values (e.g., all my children are between the ages of 0 to 30 years). Secondly, OWL 1 was mainly focused on constructs for expressing information about classes and individuals and exhibited some weakness regarding expressiveness for properties. OWL 2 offers new constructs for expressing additional restrictions on properties, new characteristics of properties, incompatibility of properties, property chains and keys.

Thirdly, while OWL 1 allows assertions that an object property is symmetric or transitive, it is impossible to assert that the property is reflexive, irreflexive or asymmetric. The OWL 2 construct ReflexiveObjectProperty allows it to be asserted that an object property expression is globally reflexive-that is, the property holds for all individuals and lastly OWL 1 is limited to cardinality constraint such as specifying that a particular parents have at least one child. Or better still we can say that a particular laptop has at least one operating system. OWL 2 however, possesses qualified cardinality restriction which is a construct that provides restriction on the number and type of instances or values of a particular property of a class. Where the term “qualified” means that they apply only to a specific type of value rather than to the

Digital Innovations
& Contemporary Research in SCIENCE, ENGINEERING & TECHNOLOGY
Vol. 6. No. 2, June, 2018

overall property. An example is to show that an instance of a class called Laptop can have at least one relations or association with instances of another class of Operating System that are of the type Windows. This entailment is added using the OWL2 construct called owl:ObjectMinCardinality. Fig. 2.1 shows the ontology file using the Manchester syntax.

```
prefix: lap:<http://www.semanticweb.org/sundaykissy/ontologies/2015/7/laptops.owl#>

ObjectPropertyRange
(lap:hasOS ObjectMinCardinality(1 lap:hasOS lap:LaptopOS))

Class: lap:MacBook

    SubClassOf:
        lap:Laptop,
        lap:hasOS exactly 1 lap:LinuxOS,
        lap:hasOS exactly 1 lap:WindowsOS
```

Fig 2.1: OWL 2 file ontology with restriction.

To show these changes pictorially, figures 2.2 and 2.3 give snap shots of the original ontology and the modified one.

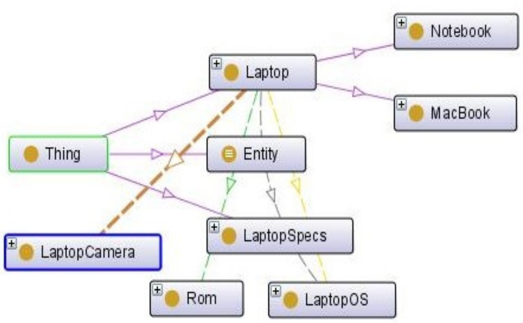

Fig 2.2 Retailer Ontology with restriction.

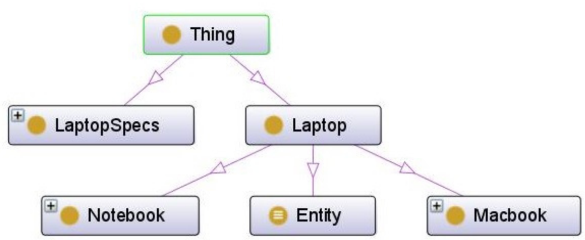

Fig 2.3 Retailer Ontology without restriction.

Qualified cardinality restriction (QCR) was also added to the core ontology. However, in this ontology, the individuals of the class of *retailers* are said to take maximum of 3 *paymentMode*. Figures 2.4 captures the ontology file written using functional syntax. In this ontology however, the OWL 2 construct that is been applied is the *owl:ObjectMaxCardinality.*

```
prefix: lap:<http://www.semanticweb.org/sundaykissy/ontologies/2015/8/untitled-ontology-63#>
ObjectPropertyRange
(lap:hasPaymentMode ObjectMaxCardinality(3 lap:hasPaymentMode lap:LaptopPaymentMode))
```

Fig 2.4: OWL 2 file of Core Ontology with restriction.

Digital Innovations
& Contemporary Research in SCIENCE, ENGINEERING & TECHNOLOGY
Vol. 6. No. 2, June, 2018

To also show these changes pictorially, figures 2.5 and 2.6 give snap shots of the original ontology and the modified one.

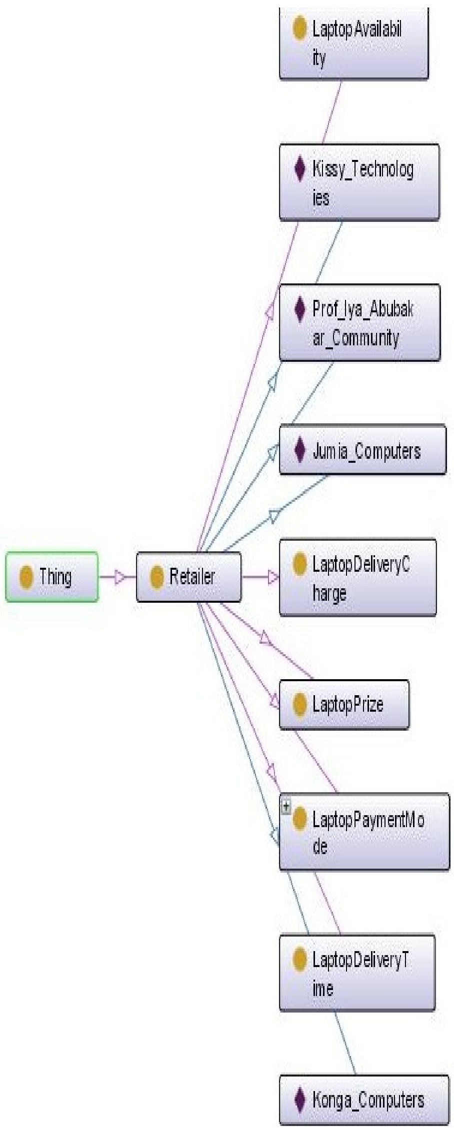

Fig 2.5: Core ontology without restriction

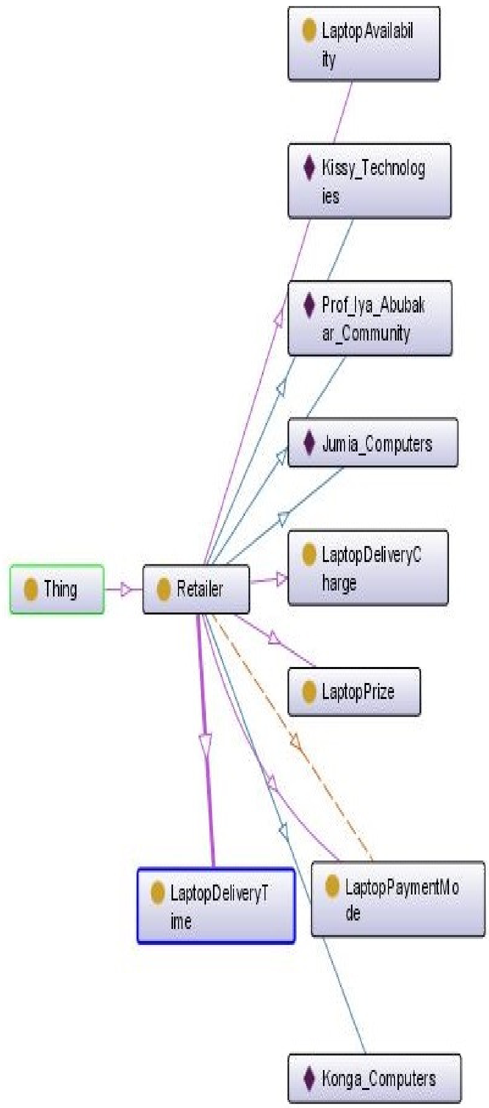

Fig 2.6: Core ontology with restriction.

3.3.1 OntoGraf

To give more support for interactively navigating the relationships of our OWL 2 ontology, we make use of the ontoGraf feature which is a plugin in our protégé ontology editing tool. It supports various layouts for automatically organizing the structure of our ontology as depicted in figure 2.7. As you can see, different relationships are supported: subclass, individual, domain/range object properties and equivalence.

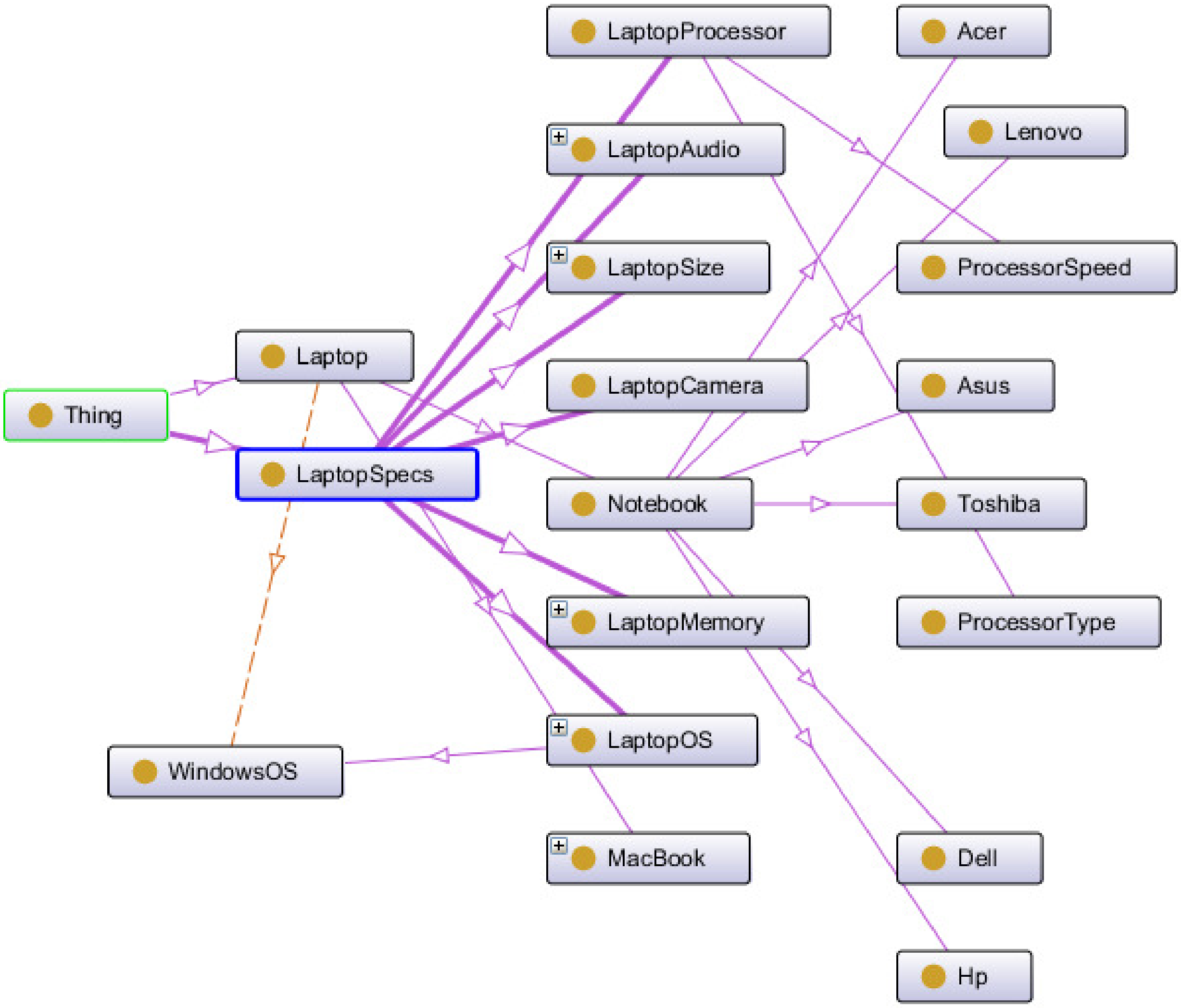

Fig: 2.7: General Structure of OWL 2 Ontology with OntoGraf

3.3.2 OntoViz

This visualization plug-in for Protege displays an ontology as a graph. Ontology classes are shown in circles, and the relationships among them are shown as arcs. The label on the arc is the name of the relationship in this case (is a). This visualization paradigm is particularly helpful for gaining more understanding of our ontology. Figure 2.8 clearly shows our ontology visualization which was achieved with the help of a sophisticated software called OntoGraph.

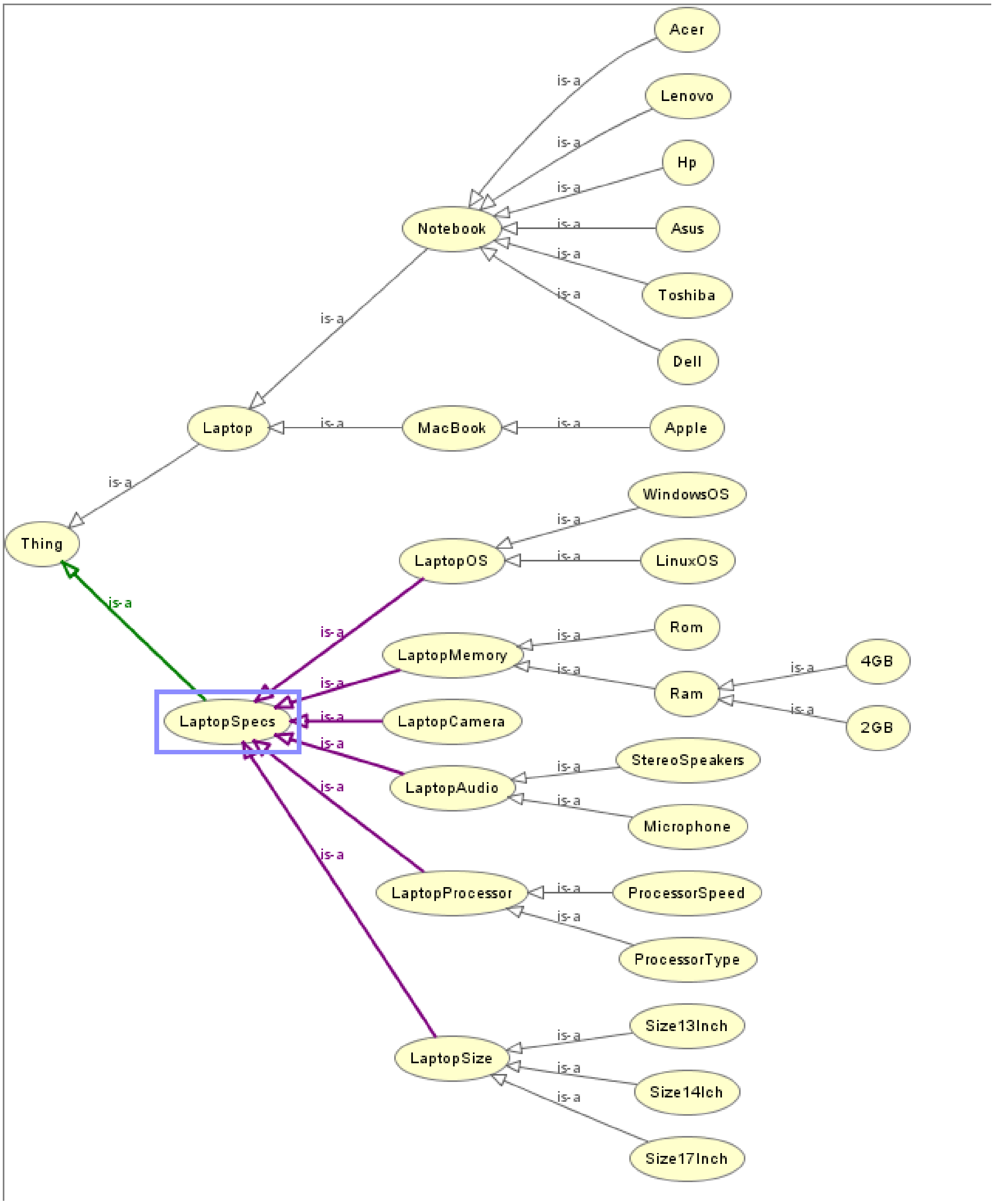

Fig 2.8: Visualizing Laptop Ontology using OntoViz

3.4 Hermit and Pellet reasoner comparison on OWL 2.
In this section, we shall briefly look at the reasoning capacity of HermiT and Pellet reasoners in inferring new fact from an OWL2 ontology. Pellet (http://clarkparsia.com/pellet) an OWL 2 DL reasoner that implements a tableaux-based decision procedure is the first sound and complete OWL 1 reasoner with extensive support for reasoning with individuals (including norminal support and conjunctive query), user defined datatypes, and debugging support for ontologies (Sirin, Parsia, Grau, Kalyanpur, & Katz, 2004). Pellet as a reasoner is a complete OWL – DL consistency checker. An OWL consistency checker takes a document as input, and returns one word being either consistent, inconsistent, or unknown. Consistency checking that Pellet provides however, ensures that an ontology does not contain any contradictory facts. But since pellet original and initial design was for OWL 1 it makes it difficult for the reasoner to infer OWL 2 ontologies successfully. For example, the Qualified Cardinality Restriction (QCR) is not present in OWL 1 and thus not supported by Pellet as shown in fig 2.9 However, an implementation of this reasoner has been successful in inferring new facts from within protégé OWL2, a java based implementation for reasoning OWL 2 is still a work in progress.

HermiT (http://hermit-reasoner.com) on the other hand is an OWL 2 reasoner both for use in a Java application through its API and as a plugin in protégé for inferring new facts. HermiT supports all features of the OWL 2 ontology language (Cuenca Grau, et al., 2008), including all OWL 2 datatypes (Motik & Horrocks, 2008), and it correctly performs both object and data property classification—reasoning tasks that are, to the best of our knowledge, not fully supported by any other OWL reasoner. In addition to these standard reasoning tasks, HermiT also supports SPARQL query answering, and it uses a range of optimizations to ensure efficient processing of real-world ontologies. Having used OWL 2 to build our ontology, in this paper we propose the use of HermiT reasoner to infer new facts from our ontology. Fig 2.8 and Fig 2.9 show the abstract view of both HermiT and Pellet reasoner with respect to OWL 2 java application.

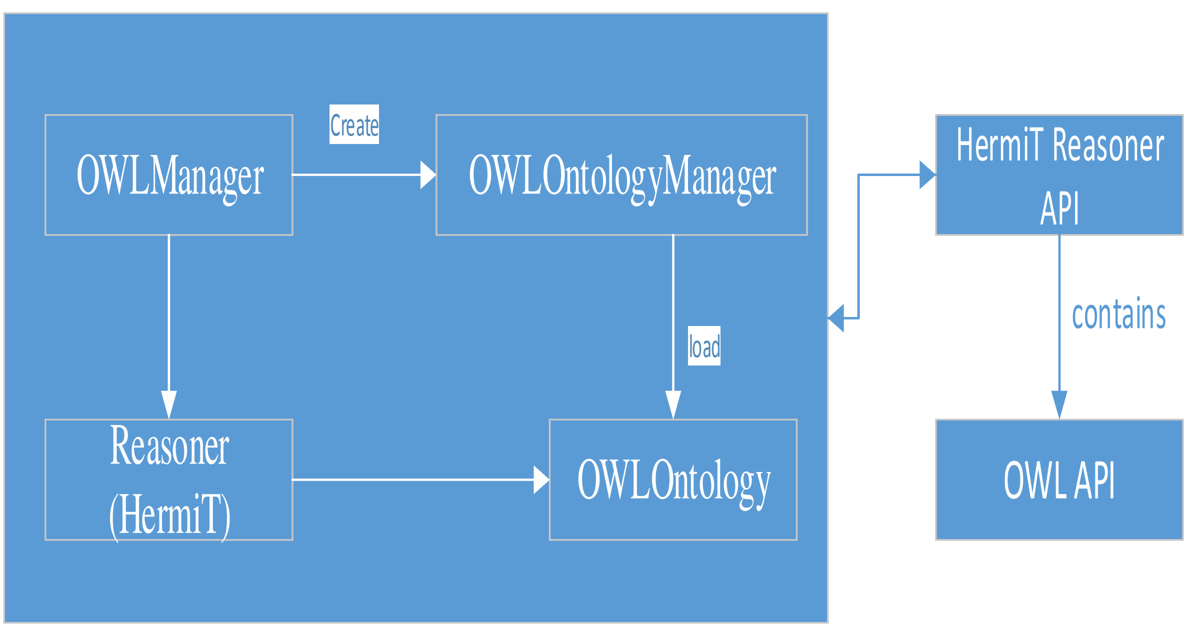

Fig 2.8 Abstract Structure of the HermiT OWLReasoner Engine.

The structure shown above presents HermiT Reasoner API as a complete reasoner which contains our OWL API necessary for communicating with our OWL 2 ontology file. The OWLManager Provides a point of convenience for creating an OWLOntologyManager with commonly required features (such as an RDF parser for example). An OWLOntologyManager manages a set of ontologies. It is the main point for creating, loading and accessing ontologies. It also manages the mapping between an ontology and its ontology document.

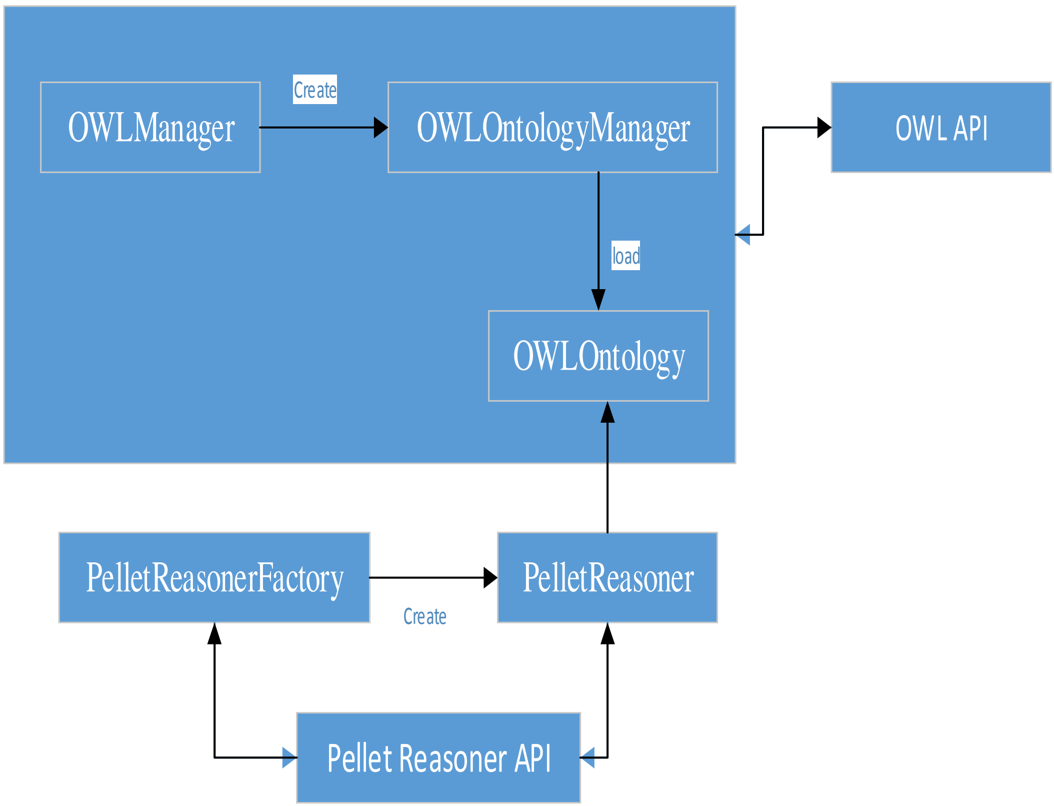

Fig 2.9 Abstract Structure of the HermiT OWLReasoner Engine.

The structure shown above presents Pellet Reasoner API which contains the PelletReasonerFactor necessary for creating the PelletReasoner which takes the OWLOntology as a parameter. OWL API is needed and necessary for communicating with our OWL 2 ontology file which has to be loaded into the java library. The OWLManager just like HermiT, provides a point of convenience for creating an OWLOntologyManager with commonly required features (such as an RDF parser for example). It also manages a set of ontologies. It is the main point for creating, loading and accessing ontologies.

4. DISCUSSION AND FINDINGS

Starting with the Retailer ontology designed in the previous section, equivalent HermiT and Pellet queries were executed on these two different ontologies using the Manchester Syntax for our class expression and the output of the result is shown in fig 2.9a and fig 2.9b respectively. Looking at Figure 2.9a, both reasoners query were executed on the unmodified or original ontology that does not have OWL2 restrictions; the result of these queries shows that they both inferred the same information. However, when the ontology was modified to in`clude more expressive construct, it was observed that as the same set of queries were being executed on the modified ontology, Pellet reasoner could not infer the instances of the ontology while HermiT reasoner could infer the ontology instances.

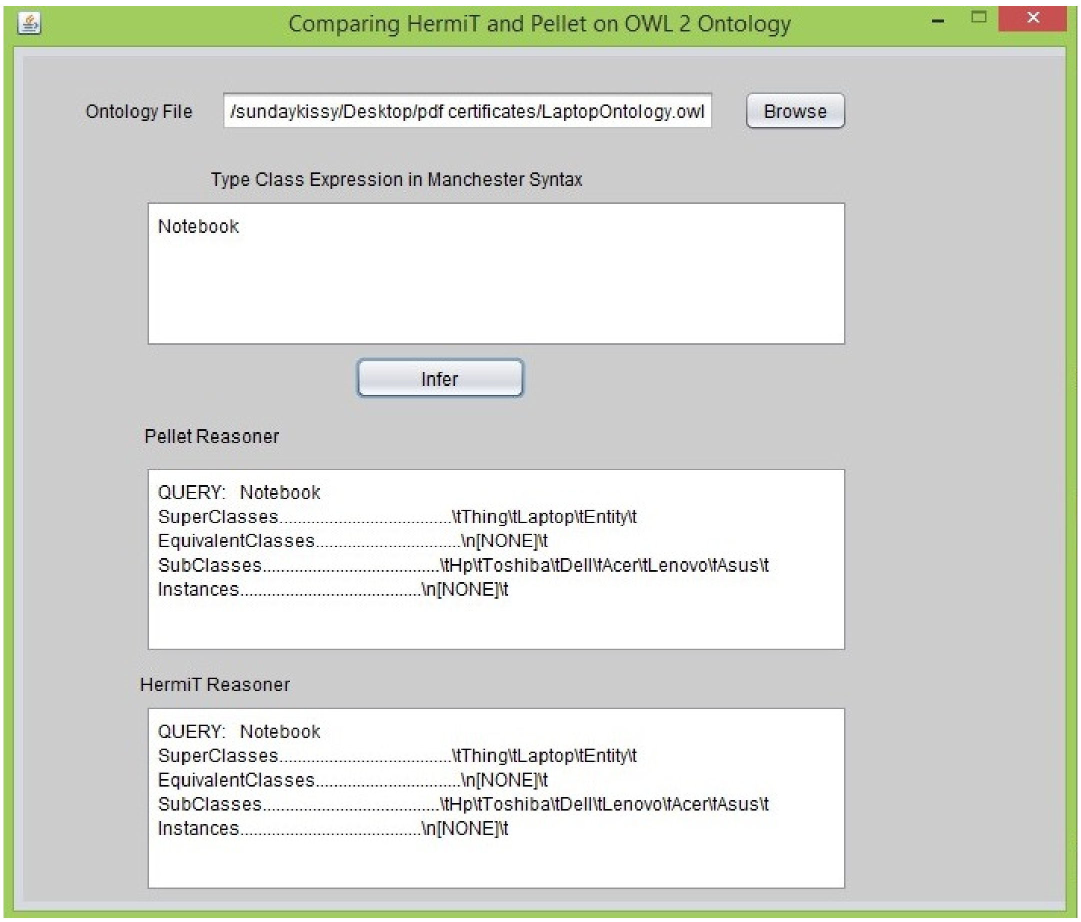

Fig 2.9a Inferring original retailer ontology

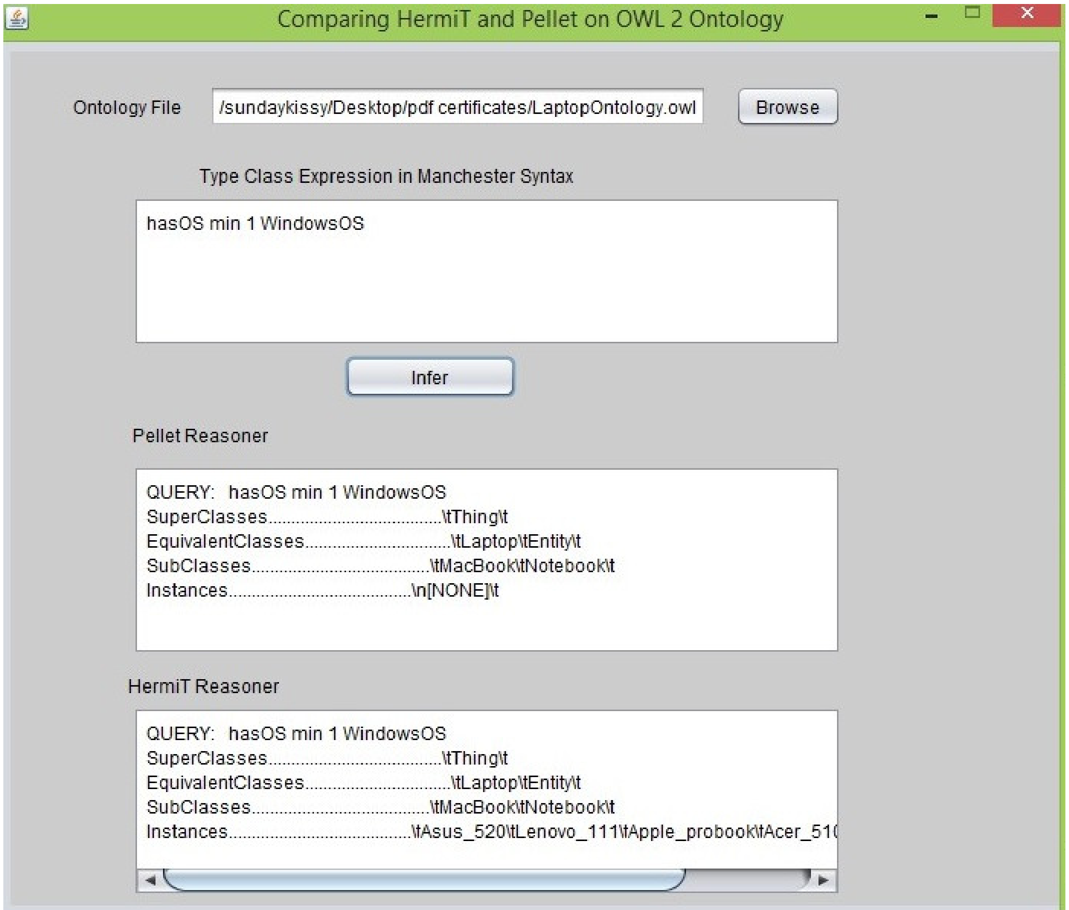

Fig. 2.9b: Inferring modified retailer ontology

REFERENCES

1. Antoniou, G., & Harmelen, F. V. (2004). *Semantic Web Primer, MIT Press, Cambridge.*
2. Berners-Lee, T. (1994). Weaving the Web. *The MIT Press, Cambridge*.
3. Cuenca Grau, B., Horrocks, I., M. B., Parsia, B., Patel-Schneider, P., & Sattler, U. (2008). OWL 2: The next step for OWL. . *Journal of Web Semantics*, 6(4), 309-322 .
4. Fensel, D., Harmelen, F. v., Horrocks, I., McGuinness, D. L., & Patel-Schneider., P. F. (2001). OIL: An ontology infrastructure for the semantic web. IEEE Intelligent Systems. 38-45.
5. Grau, B. C., Horrocksa, I., Motika, B., Parsia, B., Patel-Schneider, P., & Sattler, U. (2008). OWL2: The Next Step for OWL.
6. I., H., Patel-Schneider, P. F., & van Harmelen, F. (2003). From SHIQ and RDF to OWL: The Making of a Web Ontology Language. *Journal of Web Semantics*, 7–26.
7. Motik, B., & Horrocks, I. (2008). OWL Datatypes: Design and Implementation. In: Proc. of the 7th Int. Semantic Web Conference (ISWC), Lecture Notes in Computer Science. *International Semantic Web Conference* (pp. vol. 5318, pp. 307–322.). Springer.
8. Oyelade, O. N., Junaidu, S. B., & Obiniyi, A. A. (2014). Semantic Web Framework for E-Commerce Based on Web Ontology Language Vol 11 issue 3 No 2. *International Journal of Computer Science Issues*, 1694-0784.
9. Patel-Schneider, P. F., Hayes, P., & Horrocks, I. (2004). OWL Web Ontology Language Semantics and Abstract Syntax, W3C Recommendation.
10. Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2004). Pellet: A practical OWL-DL Reasoner.