

A Category Theoretic Model of RDF Ontology

S. Aliyu¹, S.B. Junaidu², A. F. Donfack Kana³

^{1,2,3} Department of Mathematics, Faculty of Science, Ahmadu Bello University, Zaria.

ABSTRACT

Ontology languages are used in modelling the semantics of concepts within a particular domain and the relationships between those concepts. The Semantic Web standard provides a number of modelling languages that differ in their level of expressivity and are organized in a Semantic Web Stack in such a way that each language level builds on the expressivity of the other. There are several problems when one attempts to use independently developed ontologies. When existing ontologies are adapted for new purposes it requires that certain operations are performed on them. These operations are currently performed in a semi-automated manner. This paper seeks to model categorically the syntax and semantics of RDF ontology as a step towards the formalization of ontological operations using category theory.

KEYWORDS

RDF, Ontology, Category Theory

1. INTRODUCTION

A formal representation of knowledge is based on conceptualization of the objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold them [1]. A conceptualization is an abstract, simplified view of the world that is represented for some purpose [2]. Every knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly.

An ontology is an explicit specification of a conceptualization [2]. The use of ontologies in information systems is becoming more and more popular in various fields, such as web technologies, database integration, multi agent systems, natural language processing, semantic web etc. The Semantic Web is a revolution in the World Wide Web which has gained the attention of many researchers. Semantic Web describes methods and technologies to enable machines to understand the semantics of data on the World Wide Web using Ontologies. Ontologies include computer-usable definitions of basic concepts in a domain and the relationships among them. They encode knowledge in a domain and knowledge that spans domains. In this way, knowledge reusability is promoted. The availability of machine-readable ontologies would enable automated agents and other software to access the web more intelligently. The agents would be able to perform tasks and locate related information automatically on behalf of the user [3].

It will rarely be the case that a single ontology fulfils all the needs of a particular application domain. More often, multiple ontologies are combined. This has raised the ontological

composition problems. Ontology Composition refers to those operations(e.g. instantiation, subsumption, satisfiability, alignment, merging, mapping etc.) involved when a single ontology is modified or combined with another to form another ontology. As noted by [3], ontology composition problems is widely seen as both a crucial issue for the realization of the semantic web and as one of the hardest problems to solve. Ontology composition has received wide attention in the research community in recent years [4], [5], [6], [7], [8], [9], [10] and [11].

[5] observed that another important problem to be solved while combining ontologies is the automation of the process. Techniques that rely strongly on human input are able to score better on precision, but, are less scalable and labour intensive as compared to methods that rely strongly on automation. [3] also noted that the sharing of information based on the intended semantics, is still a fundamental challenge.

It is believed that a formal view on ontologies can contribute a lot in solving ontology composition problems. As noted by [4], it is essential that the chosen formalism emphasizes relationships between things, allowing mapping in an appropriate manner, allowing coexistence of heterogeneous entities and also offering a good set of operations to put entities together.

This paper uses category theory to define a formal syntax and semantics for RDF ontologies which is the foundation on which other components of the semantic web stack is built upon.

Category theory has been successful in situations where interoperability is crucial as in formal specification of systems and in software architecture. A direct gain of this formalization is the modularization and reuse of the framework.

The rest of this paper is structured as follows: Section 2 introduces the Semantic Web Stack, Ontological languages and Category Theory. Section 3 describes the RDF Ontological model. Section 4 then describes the modelling of our RDF Ontology Categorically with examples. Section 5 introduces our interpretation model and then model the semantic of our Categorical RDF Ontology. Section 6 discusses the related works and finally, Section 7 gave a brief conclusion to this paper.

2. PRELIMINARIES

2.1 Semantic Web Stack

The Semantic Web was designed as an information space, with the goal that it should be useful not only for human-human communication, but also that machines would be able to participate and help [18]. One of the major obstacles to this is that, most information on the Web is designed for human consumption, and even if it was derived from some formally defined representation such as Relational Database or some early knowledge representation technique such as Semantic Network or frame based system, the use of the data is not evident to a robot browsing the Web. The Semantic Web provides languages for expressing information in a machine process-able form. The Semantic Web is a long-term project started by W3C with the stated purpose of realizing the idea of having data on the Web defined and linked in a way that it can be used by machines not just for display purposes but for automation, integration, and reuse of data across various applications [12]. The Semantic Web is designed to allow reasoning and inference capabilities to be added to the pure descriptions of knowledge in a domain. This includes stating

facts which can also be extended to the formation of complicated relationships. This allow intelligent software to act on this descriptive information. The Semantic Web stack as illustrated in figure 1 is a layered cake illustrating key technologies that makes semantic web vision possible.

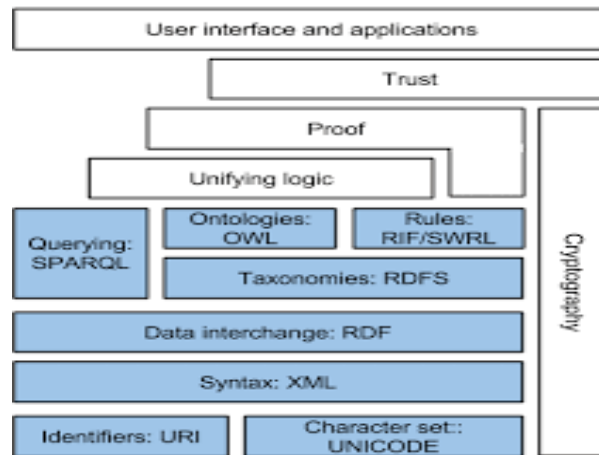


Figure 1: Semantic Web Stack (Source: <http://www1.cse.wustl.edu/>)

The semantic web stack has at its base the URI (Universal Resource Identifier), a compact string of characters used to identify or name a resource. IRI (Internationalized Resource Identifier) is a form of URI that uses characters beyond ASCII, thus becoming more useful in an international context. Also at the same level to the URI, is the Unicode which is the universal standard encoding system and provides a unified system for representing textual data. Immediately above that layer is the XML (Extensible Markup Language). XML provides a standard way to compose information so that it can be easily shared. XML allows users to add arbitrary structure to their documents but says nothing about what the structures mean. This leads us to RDF—the Resource Description Framework. The W3C developed this new logical language to facilitate interoperability of applications which generate and process machine-understandable representations of data resources on the Web. In RDF, a document makes assertions that particular things have properties (such as "is a brother of," "is wife of") with certain values. This structure turns out to be a natural way to describe the majority of data processed by machines. Within this structure, the subject and object are each identified by a Universal Resource Identifier (URI). Because RDF does not make assumptions about any particular domain, nor does it define the semantics of any domain, RDFS was developed so as to achieve that. RDFS is a language for defining the semantics of a particular domain. RDFS is RDF's vocabulary description language. RDFS is too inexpressive and cannot be used to define constraint on relationships to be other than m:n (many-to-many). Clearly, for realistic applications something better than RDFS was needed. For such reasons, the Web Ontology Language (OWL) was developed. OWL is a Knowledge Representation language proposed by the W3C as a standard to codify ontologies in a prospective Semantic Web. OWL is based on Description Logics. We can represent a knowledge domain computationally in an OWL ontology, in order to apply automated reasoning, infer knowledge, queries, classify entities against the ontology, integrate knowledge from different resources etc.

2.2 Category Theory

A Category is a collection of data that satisfy some particular properties. Category theory is the mathematical theory of structures and its greatest significance lies in its capacity to express relationships between structures. Examples of category include: a database schema which is a system of tables linked by foreign keys, an ontology representation of concepts in a particular domain.

Category Theory is designed to describe various structural concepts in a uniform way. A *category* models entities of a certain sort and the relationships between them [17]. A categorical structure is similar to a graph where the nodes are entities and the arrows are relationships. For example, figure 2 shows the relationship between a Categorical structure and a graph. Where the entities are the nodes represented as A, B, C and the relationships are the arrows represented as f, g, h.

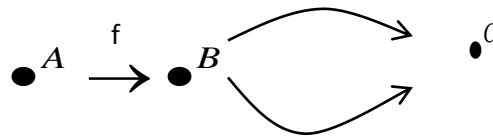


Figure 2: A Categorical Structure

For example, the statement from [17]:

"self email is an email from a person = self email is an email to a person" can be categorically modelled as:

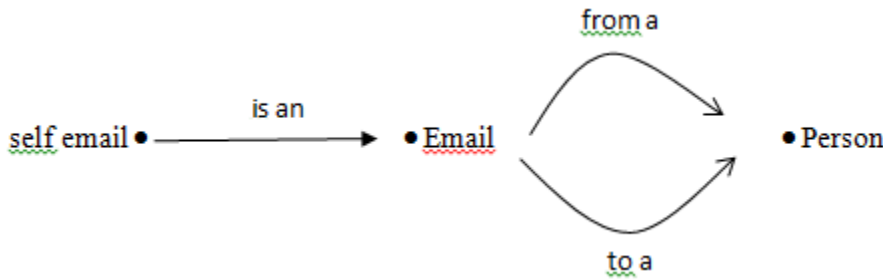


Figure 3: A Categorical Statement Model

A Categorical structure is built on the following constituents:

- a) A collection of things called *objects*, denoted by: A, B, C, \dots varying over objects.
- b) A collection of arrows called *morphisms* or *maps*, usually denoted by: f, g, h, \dots varying over morphisms.
- c) A relation on morphisms and pairs of objects, called *typing* of the morphisms. For morphism f and objects A, B , the relation is denoted $f: A \longrightarrow B$. We



also say that A is the type of f and that f is a morphism from A to B . A and B are referred to as domain and co-domain of f respectively.

For each pair of morphisms say: $f : A \longrightarrow B$ and $g : B \longrightarrow C$
 a composite map holds: $g \circ f : A \longrightarrow C$

A Categorical structure must also satisfy the following rules, laws or axioms.

Identity Laws: if $f : A \longrightarrow B$, then $1_B \circ f = f$ and $f \circ 1_A = f$

Associative law: if $f : A \longrightarrow B$ and $g : B \longrightarrow C$ and $h : C \longrightarrow D$, then $(h \circ g) \circ f = h \circ (g \circ f)$

3. RDF ONTOLOGY

The Resource Description Framework (RDF) is a framework for representing information on the Web [13]. The core structure of the abstract syntax is a set of triples, each consisting of a *subject*, a *predicate* and an *object*. A set of such triples is called an *RDF graph*. An RDF graph can be visualized as a node and directed-arc diagram in which each triple is represented as a node-arc-node link [13]. There can be three kinds of nodes in an RDF graph: IRI's, literals and blank nodes. Asserting an RDF triple says that some relationship, indicated by the predicate, holds between the resources (nodes) denoted by the subject and object. This statement corresponding to an RDF triple is known as an *RDF simple statement* [13]. An RDF ontological structure can now be defined as follows:

Definition 1 (An RDF Ontology/Graph): An RDF Ontological structure is a 3-tuple $\langle R, P, h \rangle$

where:

- R is a disjoint set of IRI's which identifies resources, literals and blank nodes.
- P is a set of properties
- h is a mapping from P into the powerset of $(R \times R)$, which is an operation that associates to each property in P its domain and range objects.

Assuming pair wise disjoint set of IRIs denoted by I , the set of blank nodes denoted by B and the set of literals denoted by L .

An RDF triple (simple statement) is a tuple $\langle s, p, o \rangle \in (I \cup B) \times I \times (I \cup B \cup L)$ and an RDF graph is a set of RDF triple [14].

RDF also provides a construct which refer to a collection of resources known as RDF containers and also make statements about statements referred to as *reification*.

This paper defines simple categorical RDF statements and handles the *container construct* with a repeated property construct i.e. using a resource to form multiple statements with the same

predicate. For reification, we will build a model of the original statement and this model will be a new resource to which additional properties can be attached.

Based on the description of a Categorical structure in Section 2, a categorical structure can now be defined as follow:

Definition 2 (A Category): A Category C is a structure (O, M, f, o, id) , where:

- O is a collection of objects.
- M is a collection of morphisms say $g \in M$ such that $g: A \longrightarrow B$ where $A, B \in O$.
- $f: M \longrightarrow O \times O$ is an operation which associates to each morphism its domain object and codomain object.
- o is an associative operation of morphism composition.
- id is a collection of identity morphism for each object of O .

Definition 2 can now be use to model ontological languages categorically.

4. CATEGORICAL RDF ONTOLOGY MODEL

Definition 3 (A Categorical RDF Ontology): A Categorical RDF Ontology C is a structure (R, P, f, o, id) , where:


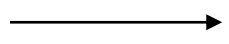
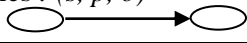
- R is a collection of resources;
- P is a collection of morphisms (e.g. $g: A \longrightarrow B$) where $g \in P, A, B \in R$
- $f: P \longrightarrow W$, where $W \subset R \times R$, f is an operation which associates to each morphism its domain object and codomain object;
- o is an associative operation of morphism composition;
- id is a collection of identity morphisms for each object of R .

From definition 3, the whole bunch of vocabulary to be used are gotten from the union of the two sets R and P . In order not to reinvent the wheel, vocabulary used in this paper will be those provided by RDF Syntax specification. The semantics to these syntax will be based on the RDF Semantics specification but the interpretation or mapping of these syntax to their intended meaning or semantics will be defined categorically in our model.

Definition 4 (Categorical RDF Triple): A Categorical RDF Triple or simple statement is a category $p: S \longrightarrow O$ where S and O are categorical objects representing resources and p is a morphism that maps from S to O .

RDF has an abstract syntax that reflects a simple graph-based data model [13]. Table 1 shows the relationship between the abstract syntax and the Categorical RDF Ontology model.

Table 1: Categorical RDF Ontology Resource, Properties and Statement description

Basic Data Model	Abstract Syntax	Categorical RDF Ontology
Resource	Labelled node with <i>URI</i> reference (<i>Subject (s)</i> or <i>Object (o)</i>) 	Objects e.g. A, B, S, O
Properties	Arc with <i>URI</i> reference (<i>p</i>) 	Morphisms e.g. <i>f, g, h, p</i>
Statement	Triples : (<i>s, p, o</i>) 	$p: S \longrightarrow O$

To illustrate our definitions, consider the following example:

Example 1. Consider a place description graph or ontology in which a particular place identified by Samaru is in a City, Zaria and Zaria is in a Country, Nigeria and Nigeria is within a particular Continent, Africa. The resource, Zaria is also related to a particular Geo Location and the Geo Location has both a latitude value and a longitude value. Zaria as a place is also associated to a population value. A graphical RDF model of a place description is pictured in Figure 3.

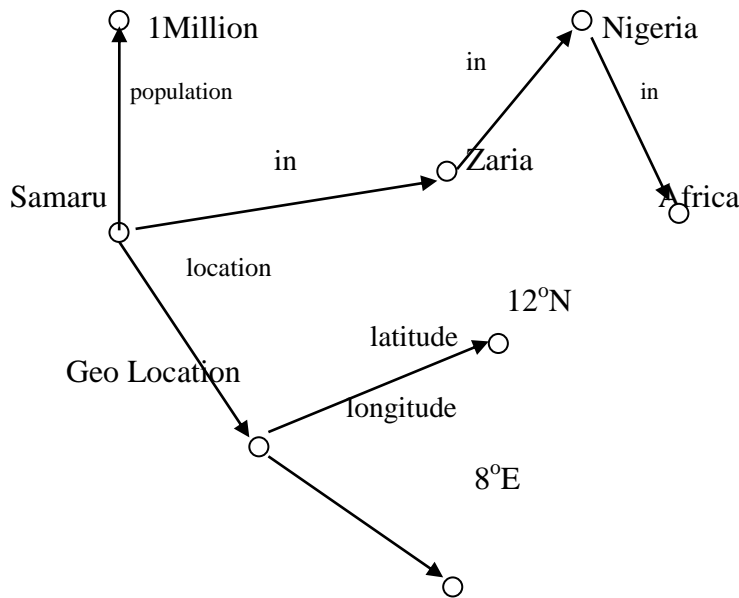


Figure 3: Graph based RDF Place Description

From figure 3, the following Categorical RDF Ontology constituents can be obtained:

- The set of resources, $R = \{Samaru, Zaria, Nigeria, Africa, Geo\ Location, 1\ Million, 8^\circ E, 12^\circ N\}$.
- The set of predicates or properties, $P = \{population, location, longitude, latitude, in\}$.

- The subset of the Cartesian product on R , represented by $W = \{ \langle \text{Samaru}, \text{Zaria} \rangle, \langle \text{Zaria}, \text{Nigeria} \rangle, \langle \text{Nigeria}, \text{Africa} \rangle, \langle \text{Samaru}, \text{Geo Location} \rangle, \langle \text{Samaru}, 1 \text{ Million} \rangle, \langle \text{Geo Location}, 8^\circ \text{E} \rangle, \langle \text{Geo Location}, 12^\circ \text{N} \rangle \}$

So, a morphism mapping from the set of properties, P , to an element in W forms an RDF statement like: *Samaru is in Zaria*. It should be noted that in forming the statements from W , it is possible that not all $\langle A, B \rangle$ can be used. Some may be meaningless in the context of the ontology domain.

5. CATEGORICAL RDF ONTOLOGY SEMANTIC MODEL

In order to describe the semantics of the Categorical RDF Ontology Model, an important construct in Category Theory known as *functor* is worth defining. [15] has defined a functor as follows.

Definition 5 (A Functor): A functor $F: C \longrightarrow D$ is a pair of functions F_0 and F_1 for which:

- if $f: A \longrightarrow B$ in C , then $F_1(f): F_0(A) \longrightarrow F_0(B)$ in D .
- For any object A of C , $F_1(\text{id}_A) = \text{id}_{F_0(A)}$.
- If $g \circ f$ is defined in C , then $F_1(g) \circ F_1(f)$ is defined in D and $F_1(g \circ f) = F_1(g) \circ F_1(f)$.

In other words, a functor is a function from one category to another. This means that, for every object or morphism in Category C there is an image, mapping, transformation or translation in the other Category D via the functor.

A functor is therefore a structure-preserving map between categories, in the same way that a homomorphism is a structure-preserving map between graphs [15].

The above definition can be illustrated in a diagrammatic format as:

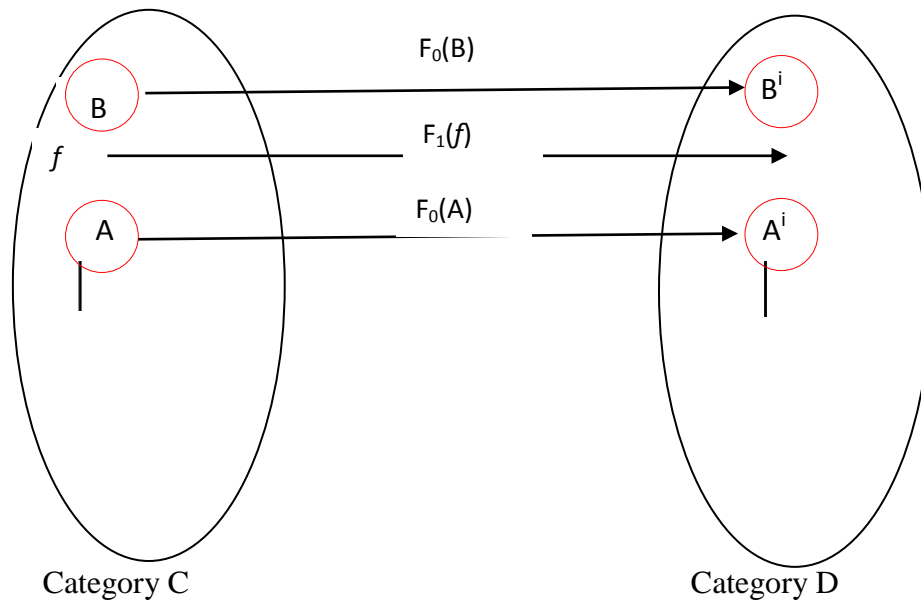


Figure 4: Functors Definition

We intend to attach meanings to the Categorical RDF ontological construct (say Category C) by mapping them to their interpretations (say Category D) via an interpretation function which is in this case a functor.

This idea is according to [16], an interpretation is a mapping from IRIs and literals into a set of interpretations, together with some constraints upon the set and the mapping. Our IRIs and literals will be our Categorical RDF Ontology while the set of interpretations will be our Categorical Interpretation Model then the mapping will be functors between the two categories.

Before defining our Interpretation Model, it is important to review the concept of subcategory as described by [15]:

Definition 6 (subcategory): A subcategory D of a category C is a category for which:

- All the objects of D are objects of C and all the arrows of D are arrows of C.
- The source and target of an arrow of D are the same as its source and target in C.
- If A is an object of D then its identity arrow id_A in C is in D.
- If $f:A \longrightarrow B$ and $g:B \longrightarrow C$ in D, then the composite (in C) $g \circ f$ is in D and is the composite in D.

Definition 6 is now used to defined our Categorical RDF Ontology Interpretation Model in Definition 7.

Definition 7 (A Categorical RDF Ontology Interpretation Model):

A Categorical RDF Ontology Interpretation Model C_1 is a 3 tuple structure $\langle R^I, P^I, F_1 \rangle$ where R^I is a collection of IRIs representing referred resources, P^I is a collection representing referred properties and F_1 is a collection of 3 operations (f_1, f_2, f_3) where these operations describes three subcategories of C_1 defined as:

- $\langle Z, P^I, f_1 \rangle$ where
 - Z is the powerset of $R^I \times R^I$ (i.e. Cartesian product of resources).
 - P^I is the collection of properties.
 - f_1 is an operation that assigns an element of P^I to an ordered pair from Z (i.e. $f_1: P^I \longrightarrow Z$).
- $\langle Y, R, f_2 \rangle$ where
 - Y is the union of referred resources and properties, $R^I \cup P^I$.
 - R^I is a collection of referred resources.
 - f_2 is an operation that maps an IRI into the union of resources and properties (i.e. $f_2: R^I \longrightarrow Y$).
- $\langle L, R, f_3 \rangle$ where
 - L represents literals where $L \subset IRIs$
 - R^I is the collection of resources.
 - f_3 is an operation that maps partially a literal into a collection of resources (i.e. $f_3: L \longrightarrow R^I$).

Now, having defined our interpretation model, the function from any Categorical RDF expression or statement into our Interpretation model describes the semantic model defined as follows:

Definition 8 (A Categorical RDF Ontology Semantic Model):

A Categorical Ontology Semantic Model is a 3-tuple $\langle C_v, C_i, F \rangle$ structures where:

- C_v is a Categorical RDF Ontology Expression.
- C_i is our Interpretation Model.
- F is our interpretation function (i.e. functor).

Let E be a Categorical RDF statement consisting of names as objects, triples or statements as binary mapping between objects and graphs as multiple triples then, the meaning can be given by the following rules defined algorithmically:

```

if (E is a literal ) then
     $F(E) \leftarrow f_3(E);$  // Interpretation of E
else if (E is a resource) then
     $F(E) \leftarrow f_2(E);$ 
else if (E is a statement with  $\langle s, p, o \rangle$ ){
    if(  $(F(p) \in P) \ \&\& \ (\langle F(s), F(o) \rangle \in f_1(F(p)))$ );
         $(F(E) \leftarrow true );$ 
    else
         $(F(E) \leftarrow false );$ 
    }
else // i.e. when we have multiple triples (say in graph, G) {
    foreach  $E^t$  in G { // where  $E^t \in G$ 
        if( $F(E^t) = false$  )
            then  $F(E) = false$ ;
        else
             $F(E) = true;$ 
        }
    }

```

Note that this rules changes as the expressivity of our ontology language changes.

6. RELATED WORK

Several other approaches and techniques exist for the effective and efficient management of ontologies. According to [5], most of these techniques are semi-automatic approaches towards ontological operations and are focused at constructing software tools that can be used to combine independently developed ontologies. In contrast, the approach taken in this paper is to develop a systematic mathematical theory for ontological operations management. Other works that have also considered a mathematical approach include the work of [5] which described an algebra for composing ontologies by defining a recursive finite typing of RDF collections for expressing ontologies expressed in heterogeneous languages, then constructing a well-founded algebraic scheme over a universe of ontologies that respects well known algebraic operations and identities.

The semantics of his model was defined as well founded set in ZF set theory. [5] work has the limitation of considering only the RDF level in the semantic web stack. [4], viewed ontology as a structural construct where operations between more than one ontologies are performed structurally with no consideration of the semantics of the data involved.

7. CONCLUSION

This paper has modelled the RDF Ontology using an algebraic structure, Category Theory. We reviewed related work in which most are focused on the practical implementation of software applications to manage ontological operations with limited theoretical or conceptual understanding underlying such operation. As a result, they fail to precisely state the details of the operations they describe. Hence, we developed a theoretical basis to define precisely ontological operations by first modelling its base (RDF) constructs. Reification and Container constructs are not left out in our model. we can capture most of the RDF modelling constructs categorically as we intend to build upwards along the semantic web stack to fully capture the requirements of the Semantic Web and efficiently managing its information represented as ontologies both syntactically and semantically. Thus, with a fully modelled ontology, ontological statement can be reduced to its mathematical equivalent.

REFERENCES

- [1] Genesereth, M. R., & Nilsson, N. J. (1987). *Logical Foundations of Artificial Intelligence*. San Mateo: CA: Morgan Kaufmann Publishers.
- [2] Gruber, T. R. (1993). *A Translation Approach to Portable Ontology Specifications*. Stanford, California: Knowledge Systems Laboratory, Computer Science Department, Stanford University.
- [3] Antoniou, G. (2008). *A Semantic Web Primer*. Massachusetts Institute of Technology.
- [4] Isabel Cafezeiro, E. H. (2007). Semantic Interoperability via Category Theory.
- [5] Kaushik, S. W. (2006). An Algebra for Composing Ontologies. *Proceedings of the 4th International Conference on Formal Ontology in Information Systems* (pp. 269-275). Baltimore, Maryland, USA: Amsterdam Ios Press.
- [6] Klein, M. (2001). Combining and relating ontologies: an analysis of problems and solutions. *Proceedings of the 17th International Joint Conference on Artificial Intelligence* (pp. 53-62). Seattle: Workshop: Ontologies and Information Sharing.
- [7] Lakshmi Tulasi et.al. (2014). Survey on Techniques for Ontology Interoperability in Semantic. *Global Journal of Computer Science and Technology* , 3-5.
- [8] Zimmermann, A. K. (2006). Formalizing Ontology Alignment and its Operations with Category Theory. *Proceedings of the 4th International Conference on formal Ontology in Information Systems* (pp. 277-288). Baltimore, Maryland, USA: IOS Press.
- [9] Euzenat, J. (2008). Algebras of ontology alignment relations. *Proceedings of the 7th International Semantic Web Conference*, (pp. 387-402). Karlsruhe, Germany.
- [10] Mitra, P. a. (2004). An Ontology Composition Algebra. *International Handbooks on Information System, SpringerVerlag* , 93-117.

- [11] Mocan, A. C. (2006). Formal Model for Ontology Mapping Creation. *Proceedings of the 5th International Semantic Web Conference* (pp. 459-472). Athens, USA: Springer.
- [12] Herman, I. (2013). *Semantic Web Activity Statement*. <http://www.w3.org/2001/sw/Activity>.
- [13] Richard, C. (2014, February 25). *RDF 1.1 Concepts and Abstract Syntax*. Retrieved January 20, 2015, from <http://www.w3.org/TR/rdf11-concepts/>
- [14] Olaf Harfig, B. T. (2014). Foundations of an Alternative Approach to Reification in RDF. *arXiv:1406.3399v1.[cs.DB]*, 3.
- [15] Micheal Barr, C. W. (1998). *Category Theory for Computing Science*. McGill University.
- [16] Patrick J. Hayes, P. F.-S. (2014, February 25). *RDF 1.1 Semantics*. Retrieved January 20, 2015, from W3C: <http://www.w3.org/TR/rdf11-mt/>
- [17] Spivak, David I. Categorical Databases. [Online] Mathematics Department, Massachusetts Institute of Technology, 13-01-2012. [Cited:1-04-2013.] <http://math.mit.edu/~dspivak/informatics/talks/CTDBIntroductoryTalk>.
- [18] Berners-Lee, Tim. Semantic Web Roadmap. [Online] September 1998. [Cited: 30 January 2015.] <http://www.w3.org/DesignIssues/Semantic.html>.