Knowledge Graph Syntax Validation and Visual Navigation for Developing Intelligent Systems

by
Claude Asamoah

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Professional Studies
in Computing

at

School of Computer Science and Information Systems
Pace University

May 2016

We hereby certify that this dissertation, submitted by **Claude Asamoah** satisfies the dissertation requirements for the degree of Doctor of Professional Studies in Computing and has been approved.

Lixin Tao — 5/11/16

Dr. Lixin Tao — Date
Chairperson of Dissertation Committee

Ronald Frank — 5/11/16

Dr. Ronald Frank — Date
Dissertation Committee Member

Meikang Qiu — 5/11/16

Dr. Meikang Qiu — Date
Dissertation Committee Member

Seidenberg School of Computer Science and Information Systems

Pace University

Abstract

Knowledge Graph Syntax Validation and Visual Navigation for Developing Intelligent Systems

by
Claude Asamoah

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Professional Studies
in Computing

May 2016

Intelligent systems depend on effective knowledge representation and knowledge-based decision-making. While OWL is the dominant industry standard for knowledge representation, it has some limitations which include the lack of support for custom relations, it's reliance on the single "is-a" relation, and the emulation of other relations via complex object and data properties. Pace University has extended OWL to support knowledge graph as a replacement to better support knowledge representation and decision-making. One of the challenges is how to better support domain experts to create knowledge graphs and verify their correctness. Since a real-life knowledge graph can easily contain hundreds or thousands of classes with complex inter-relations, it is a major challenge for domain experts to review and validate their knowledge representation, and hard for application developers to fully understand the complex relations among the classes.

This research contributes a knowledge graph syntax validation algorithm and two knowledge graph visualization tools. Use cases such as the Cyber Security Communications Facilitator are used to verify the correctness and effectiveness of the contributed solutions.

Acknowledgements

I will like to thank my wife Betty Asamoah for supporting me in my quest to reach the apex of my educational aspiration. I also want to thank my children Brandon Asamoah, Ama Asamoah, and Alex Smith for their continued support in this endeavor. My special thanks to Dr. Lixin Tao, my academic advisor and Chairperson, Computer Science Department, Westchester, Chair of Ph.D. in Computer Science Program, Chair of Doctorate Professional Studies (DPS) in Computing Program for believing in me and guiding me through the completion of this Dissertation. Finally, I thank the Almighty God for giving me the strength to reach this great height of academic accomplishment.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Opportunities and Challenges in Knowledge-Based Decision-making

The Semantic Web has extended knowledge representation which was one of the goals of Artificial Intelligence (AI). AI started with pioneers such as Alan Turing (1912-1954) whose 1950 paper "Computing Machinery and Intelligence" is one of the most frequently cited in modern philosophical literature. His work is regarded by many scholars as the foundation of computer science and of the artificial intelligence program [3]. In the late 1950s and early 1960's notables such as Alan Turing, Marvin Minsky, John McCarthy and Allen Newell thought that computers that could "think" as humans do were just around the corner [4]. As cited by Harry Haplin [18], the goal of AI as stated by John McCarthy at the 1956 Dartmouth Conference is "the study to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it" [50]. The vision that machines could do practically anything humans can do did not materialize as envisaged by AI pioneers. Although AI had done well in "tightly-constrained domains," extending this ability was not sustainable [18]. Knowledge transfer that AI was supposed to address as creating a database for all the knowledge of the world did not materialize and became increasingly evident that it was rather going to create a virtual Tower of Babel of knowledge. Even within a specific knowledge representation domain such as semantic networks, it was established that a principal element such as a 'link' was interpreted in many different ways [18]. As such, Knowledge Representations were in-

accurate in determining the represented knowledge such as the use of first-order predicate logic, which was analogous to most of the knowledge representation systems used during that time [18]. Common-sense knowledge was formalized by researchers but their approach never merged into a universal platform for representing all knowledge, as stated by the influential Brachman-Smith survey [18].

Good decision-making and processing their results in an optimal manner is the consequence of quality Knowledge Representation (KR). Intelligence is "defined as the ability of a system to act appropriately in an uncertain environment, where appropriate action is that which increases the probability of success, and success is the achievement of behavioral sub goals that support the system's ultimate goal" [23]. Countless opportunities exist in knowledge based decision-making. For example, Web Mining is the extraction of pertinent data from distributed websites across the Internet. The Internet may be viewed as a huge database consisting of disparate and distributed data and at times there may be a need to traverse these various website collecting and collating pertinent data about a particular subject using agents or crawlers. Before the advent of Knowledge Based decision-making mechanism such as the Semantic Web, documents on the web contained a lot of information for computers to present them but were not understood by them. Tremendous opportunities exist for computers to understand some of the information embedded in the web documents and act upon them to the benefit of web users. The Semantic Web as described by its inventor Tim Berners-Lee as "an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [20]. The opportunity that Semantic Web brings by enabling machines to understand data and make decisions on

them will increase the viability of the web as well as present a means of doing complex and precision based activities to benefit mankind as we enter the age of Internet of Things (IoT) whereby persons, appliances, gadgets, and computers can communicate in unison.

OWL (Web Ontology Language) is the industry standard language for knowledge representation. There are tools that make it easier to build ontologies such as the Stanford Protégé. Stanford Protégé is used to create the OWL document in an OWL/XML, RDF/XML, and other formats. The RDF/XML based OWL document relates two classes together using the "is-a" relation. The "is-a" relation is currently the only first class relation used by OWL to relate two classes without user declaration. The use of this single relation "is-a" limits the ability and flexibility to relate two classes in profound ways by a subject domain expert who may want to express the relations between two classes in specific ways using custom relations. Though you can link two individual entities in a triple format via object properties and also link a data value to an individual entity via data properties, this approach of defining the relations of two classes using properties and their respective data properties if warranted via restrictions, is more complex than using the parsimonious approach of custom relations of Knowledge Graphs.

Another challenge is how to use OWL to represent knowledge in visual knowledge navigation and review. Although OWLViz, an add-on in Protégé, can graphically display the relations diagrams between classes, if the ontology is large, it becomes cumbersome to visualize all the relations in its entirety in the Knowledge Graph (KG) and therefore there is need for a visual navigation mechanism whereby all the entities, classes and relations, can be navigated easily.

1.2 Knowledge Representation Alternatives

1.2.1 Rule Based Approach

Judgmental knowledge is allowed by Rule-based inference systems about a specific problem domain to be represented as a collection of discrete rules. Each rule states that if certain assumptions are understood, then certain conclusions can be inferred [27]. W3C chartered the Rule Interchange Format (RIF) Working Group and tasked them to produce extensions in addition to a core rule language which in combination, allow rules to be translated between rule languages and subsequently between rule systems [34]. The RIF Working Group is challenged to amalgamate the needs of a diverse community including business rules and semantic web [34]. The Semantic Web Rule Language (SWRL) includes a high-level abstract syntax for Horn-like rules in both the OWL DL and OWL Lite sublanguages of OWL [35]. Rules are modes for representing knowledge and most often goes beyond OWL1 and are typically conditional statements in the Semantic Web for example the *if then clauses* [36]. In addition, rules expand the expressive power of SWRL. Rules are straight forward and mainly correspond to traditional operations available in major programming languages. Examples are Comparisons, Mathematical transformations, and modifiers. One of the popular rules used in the semantic web is the Jena Rule which includes a list of body terms or premises (the "if" clause) and a list of head terms or conclusions (the "then" clause) [36].

1.2.2 Web Ontology Language – OWL

According to Tom Gruber, an "ontology is a specification of a conceptualization" [1]. Ontologies enable the structuring of data in a hierarchical form. Recent work in Artificial

Intelligence (AI) is exploring the use of formal ontologies as a way of specifying content-specific agreements for the sharing and re-use of knowledge among software entities [11]. Web Ontology Language (OWL) is part of the Semantic Web which extends the current Web by extending current semantics to it [12]. OWL is one of the most used languages for creating ontologies which is the latest recommendation of W3C and is based on the RDF schema.

OWL = RDF schema + new constructs for expressiveness [13].

RDF is the basic block for supporting the Semantic Web and is all about vocabulary or metadata and supported by W3C. It is structured, machine readable, and capable of describing any resource independent of any domain [13].

OWL has three sublanguages OWL Lite, OWL DL, and OWL Full.

OWL Lite supports those users primarily needing a classification hierarchy and simple constraints [22].

OWL DL supports those users who want the maximum expressiveness while retaining computational completeness [22].

OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees [22].

Ontology is a vehicle to capture knowledge about a particular domain. It describes the relations between the entities within that domain. Web Ontology Language, OWL, is one of the most used languages for creating ontology which is the latest recommendation of W3C. The OWL consists of individuals, properties, and Classes.

Individuals are instance of a class. For example, John is an individual or instance of a Student class or USA is an individual or instance of a Country class.

Properties are links that relates two individuals together. For Example, *John livesIn Boston*. Properties have many characteristics such as inverse, transitive, asymmetric, or symmetric.

Classes: can be described as a set that has individuals as members. Examples of a class are Country and Car.

OWL uses the Triple, as agreed by W3C, to describe the relations between two entities. Figure 1 depicts this relation.

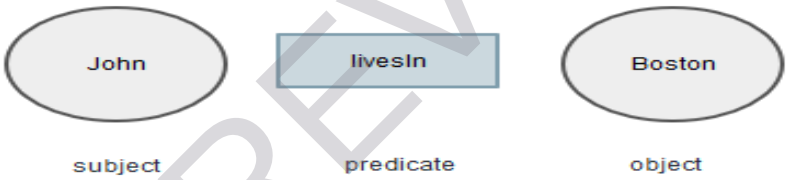

Figure 1: Triple Example

The most popular tool for creating OWL document is the Protégé. There are many versions of Protégé such as the Stanford University Protégé. The Stanford University Protégé supports only one first class relation, the "is-a" relation. For example, an *Audi is-a car*. In this triple, "Audi" is the object, "is-a" is the predicate, and "car" is the subject. With "is-a" as the only relation to link two classes, properties may have to be used to

capture a more vivid knowledge of the domain of interest. Figure 2 depicts an ontology created with the “is-a” relation of a class “Car” and its subclasses.

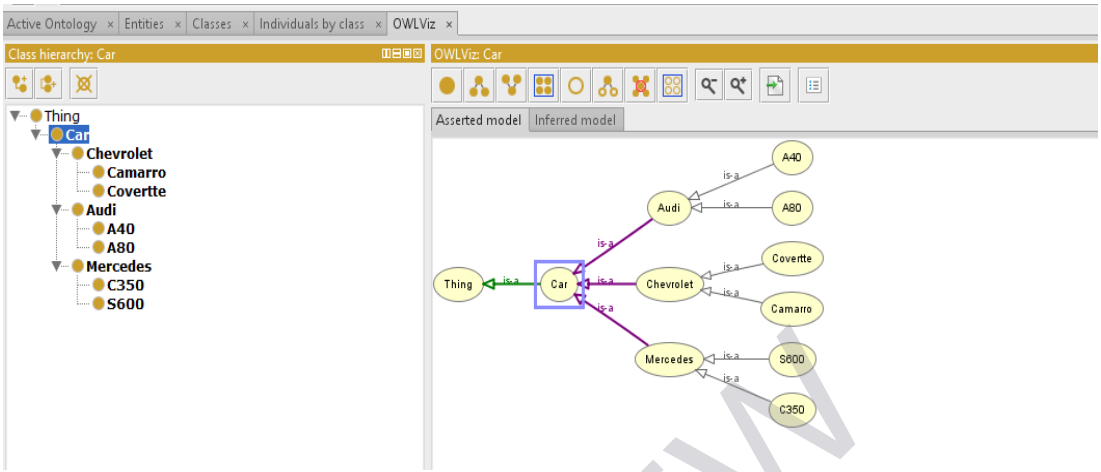

Figure 2: Car Relations Using “is-a” Predicate

It is clear that if there is a way to use more types of predicates to relate two classes, the capture of the knowledge of the domain class will be more vivid. This limitation necessitated the design and development of the Pace University Extended Protégé that has the ability to create custom relations to link two classes of a domain in question.

1.2.3 Knowledge Graph

Pace University Extended Protégé is a tool for extensible knowledge representation supporting custom relations. It is a tool for domain experts to describe and validate knowledge. Also, it is able to drive knowledge based decision-making and introduces minimal syntax extension to OWL so it can benefit from existing tools for OWL. It has the ability to relate two classes using various custom relations. Figure 3 depicts a KG document.

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rel "http://www.pace.edu/rel-syntax-ns#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<rdf:RDF xmlns="http://www.pace.edu/body-73#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:rel="http://www.pace.edu/rel-syntax-ns#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
    <owl:Ontology rdf:about="http://www.pace.edu/body"/>
    <!--
    // Relations
     -->
        <rel:NewRelation rdf:about="http://www.pace.edu/body#partOf"/>
        <rel:NewRelation rdf:about="http://www.pace.edu/body#include"/>
    <!--
    //
    // Classes
    //
     -->
    <!-- http://www.pace.edu/body#finger -->
    <owl:Class rdf:about="http://www.pace.edu/body#finger">
        <rel:partOf rdf:resource="http://www.pace.edu/body#hand"/>
    </owl:Class>
    <!-- http://www.pace.edu/body#hand -->
    <owl:Class rdf:about="http://www.pace.edu/body#hand">
        <rel:include rdf:resource="http://www.pace.edu/body#finger"/>
    </owl:Class>
</rdf:RDF>
<!-- Generated by the OWL API (version 3.5.1) http://owlapi.sourceforge.net -->
```

Figure 3: RDF/XML OWL Document Example

Figure 4 below depicts a "State" knowledge graph with custom relations or predicates viewed via OWLViz, a plugin for the Protégé tool.

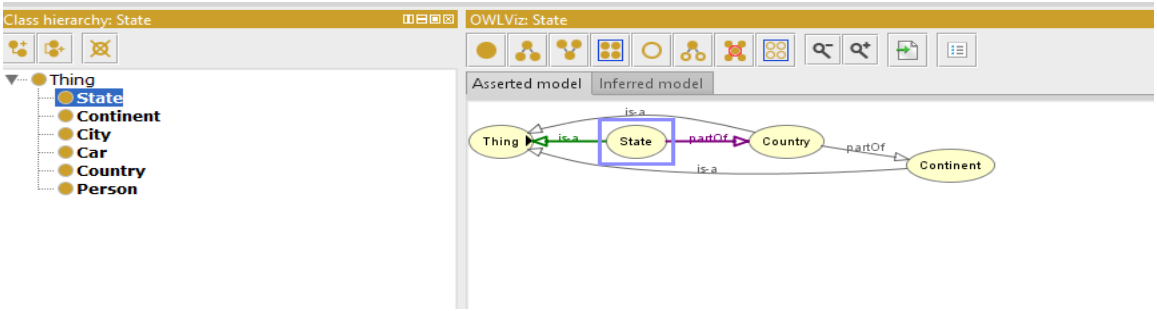

Figure 4: State Relation Using "partOf" Custom Relation

It will be more evident that if we add more custom relations to the State class, a more vivid knowledge description of the domain is presented as shown in Figure 5.

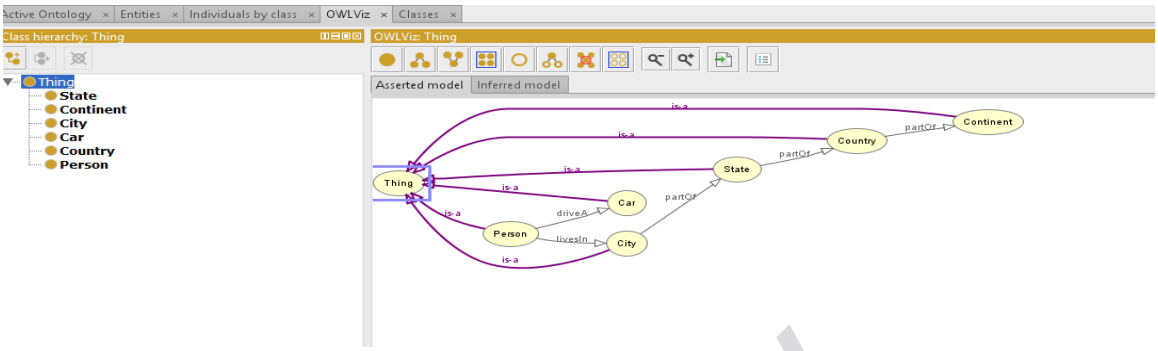

Figure 5: State Relations with More Custom Relation and Related Classes

Pace University knowledge graphs aid domain experts to design Knowledge Representation for Intelligent Systems more effectively than OWL in that custom relations can be created and used to relate classes of concepts in more vivid and expressive ways. When the KG is large, it is difficult to check for the correctness and to navigate each class relation in an efficient manner prompting the need of a visual navigation system that is easy and flexible. Before the KG can be navigated it has to be syntax validated to prevent unexpected eventualities and application system failure due to the KG not been well-formed. It is therefore of paramount importance that a syntax validation system be created to validate any KG created with customs relations. The syntax validation is a method designed in Pace Jena which is used to validate the KG before employing other methods designed in Pace Jena to parse the classes and relations of the KG and present a visual navigation capability. Though Pace Jena can parse and print out the class relations through its API, it was not intuitive as a tool in terms of flexibility, repeatability, and portability. It was therefore necessary to design a visual

navigation application facilitated by methods created in Pace Jena to display any KG currently. Pace Jena was developed by Pace University and it has the capability to parse and print out entities such as classes, custom relations, object properties, data properties, individual classes and other entities from KGs. It can also parse the KG into subject, predicate, and object in the Triple format. The subject and objects are classes that are related by customs relations or predicates. Before the Pace Jena parses the KG, it is prudent that it must first validate the KG and confirm that it is well-formed and OWL syntactically correct. The "DomParse.java" written by Dr Lixin Tao of Pace University [21] as a XML Validator is slightly modified to accommodate the ".owl" extension (which is the extension of the KG) in addition to the existing ".xml", "xsd", and "dtd" extensions. The Dom Parser will be used to validate the KG in conjunction with a designed Pace RDF/XML centric schema. Before the validation of the KG is realized, some challenges have to be overcome before the validation is possible. Since the KG is in RDF/XML format and includes custom relations, using the W3C XML OWL Schema.xsd to validate the KG with ".owl" extension is a challenge in that the W3C XML OWL Schema.xsd supports only owl documents serialized in OWL/XML format and will not work with RDF/XML serialized ".owl" extension KG. Using other RDF/XML Schemas from third parties on the web will not work either since the few encountered via research will not recognize the namespace of Pace University custom relations. The generated RDF/XML serialized KG has four distinct namespaces. They are "rdf", "rdfs", "owl", and "rel" namespaces respectively. Since an ".xsd" schema can only have one target Namespace, a unique method needed to be devised for all four namespaces to work together as a single schema. It was discovered that a separate

schema needs to be created for each namespace. The main schema will be the "rdf" namespace called main.xsd and will import the rel.xsd, rdfs.xsd, and owl.xsd schemas. The rel.xsd schema will import the main.xsd, the rdfs.xsd, and the owl.xsd schemas, the owl.xsd schema will import the main.xsd, rdfs.xsd, and rel.xsd schemas, and subsequently, the rdfs.xsd will import the rel.xsd, main.xsd, and owl.xsd schemas. By this configuration, all four namespaces persist in each individual schema even though only one target Namespace is permitted in a schema. Once the four namespaces are created and working together as one unit via the main schema, the next challenge is how to make the schema generic enough that it can validate any KG with custom relations. It is worthy to note that for future KGs created with custom relations by outside parties to be syntax validated successfully, they must use the "rel" namespace created by Pace University for their custom relation elements names. The validation of any KG will require that its relation elements be declared in the Pace KG schema. To solve this problem, a custom Pace Knowledge Graph Syntax Validator (KGSV) was designed to validate any inputted KG with custom relations before it is parsed by Pace Jena to facilitate visual navigation capabilities to the Pace KG. The designed Pace KGSV will first determine if the relation element in the KG have been declared in the Pace KG schema. If they are declared, the algorithm moves to validate the KG. If there are new elements that are yet to be validated, it employs the Multiple Pass algorithm to update the Pace KG schema by declaring the new relations elements in Pace KG schema before validating the KG. During the research to find a solution to syntax validate KGs, it was deemed that the designed Pace University's Syntax Validator should be able to syntax validate a subset of RDF/XML serialized documents with or without custom relations since there is no suitable

RDF/XML centric syntax validator. This effort led to the addition of two more namespaces, the “pace.xsd”, and the “xml.xsd” schemas to facilitate the syntax validation of RDF/XML serialized documents inclusive with object and data properties. Three functions were created in Pace Jena to facilitate visual navigation capabilities for the KG. The first function will load and return an array list with all unique classes in the KG. The second function will load and return all unique relations in the KG, and the third function takes a class name and a relation name as arguments and returns all instances of the triple namely a class instance as the subject, the custom relation as the predicate, and the related class instance as the object. The Pace Jena has been extended to provide visual navigation capabilities for the knowledge graph that can display all Classes and Relations of the KG and provide navigation in Web and Application models based on any inputted custom relation laden KG.

1.3 Problem Statement

Pace University knowledge graphs enable domain experts to codify their domain knowledge more effectively for driving knowledge-based decision-making. Since the authors can freely introduce new custom relations with various mathematical properties and use them in the same document, knowledge graph breaks the limitation of XML syntax so the standard XML syntax validator cannot validate knowledge graphs. Since a real-life knowledge graph can easily contain hundreds or thousands of classes with complex inter-relations, it is a major challenge for domain experts to review and validate their knowledge representation, and hard for application developers to fully understand the complex relations among the classes.