

# A Review of Ontology Development Aspects

Nur Liyana Law Mohd Firdaus Law<sup>1</sup>, Moamin A.  
Mahmoud<sup>2</sup>, Alicia Y.C. Tang<sup>3</sup>  
Institute of Informatics and Computing in Energy  
Universiti Tenaga Nasional, Kajang, Malaysia

Fung-Cheng Lim<sup>4</sup>, Hairoladenan Kasim<sup>5</sup>  
College of Computing and Informatics  
Universiti Tenaga Nasional  
Kajang, Malaysia

Marini Othman<sup>6</sup>  
Institute of Informatics and Computing in Energy  
Universiti Tenaga Nasional  
Kajang, Malaysia

Christine Yong<sup>7</sup>  
ICT Division  
Tenaga Nasional Berhad  
Kuala Lumpur, Malaysia

**Abstract**—Although it is widely recognized that ontology is the main approach towards semantic interoperability among information systems and services, the understanding of ontology aspects among researchers is limited. To provide a clear insight to this problem and support researchers, we need a background understanding of various aspects related to ontology. Consequently, in this paper, a comprehensive review is conducted to map the literature studies to a coherent taxonomy. These include the benefits of ontology, types of ontology, application domains, development platforms, languages, tools, and methodologies. The paper also discusses the concept of ontology, semantic Web, and its contribution to several research fields such as Artificial Intelligence, Library Science and shared knowledge. The fundamentals of ontology presented in this paper can benefit readers who wish to embark in ontology-based research and applications development.

**Keywords**—Component; ontology; semantics web; artificial intelligence

## I. INTRODUCTION

According to a book titled “Birth of a new science: the history of ontology from Suarez to Kant” [1], an Ontology is seen as an inter-discipline that involved two parts that are philosophy and science. Furthermore, the point of discipline is considered as the issue of foundations of science and not a discipline that exist separately. It is also considered as independent from other scientific discipline and branches of philosophy. Aside from science philosophy, the term is derived from general structure of the world [1]. To understand more, the structure of ontology is obtained from the knowledge that has been embodied into other disciplines. However, based on an article [2], it is said that ontology is the most comprehensive of all sciences that covers everything that exists from philosophy’s perspectives. Similarly, it is also applied to information science’s point of view with difference in essential

definition ontology. Meanwhile from psychology’s perspective, the ontology is uncommon to be used as the psychologists are more interested in studying on how people develop concepts and enhancing it from time to time.

Another article by N. Guarino, D. Oberle, and S. Staab [3] mentioned that ontology is mostly used with different meaning, depending on the communities that are utilizing the ontology. From a philosophy discipline, ontology deals with the nature and structure of “reality”. Whereas in computer science discipline, it holds a unique kind of information of computational or object artifact as a mean to formal model of the system’s structure; for example, a system able to become a company with all the employees and their relationship with one another.

As the term getting more known towards communities, various definitions were developed. The word ontology was taken from Philosophy, where it means a systematic explanation of being. In the last decade, the word ontology began used in artificial intelligence in the 1980s, and is now frequently used by computing and information science audiences. A few years later, Gruber [4] defined ontology as “an explicit specification of a conceptualization”. This definition became the most quoted in literature and by the ontology community. One other definition is from Diana Man [5], he defined ontology as “consists set of concepts within a domain including the relationship between the concepts that considered as a formal representation of knowledge in computer science”. Another article [6] defined ontology as “common vocabulary for researchers who are in need to share information in a domain”.

The fundamentals of ontology presented in this paper can benefit readers who wish to embark in ontology-based research and applications development. It maps the current literature studies to a coherent taxonomy as shown in Fig. 1.

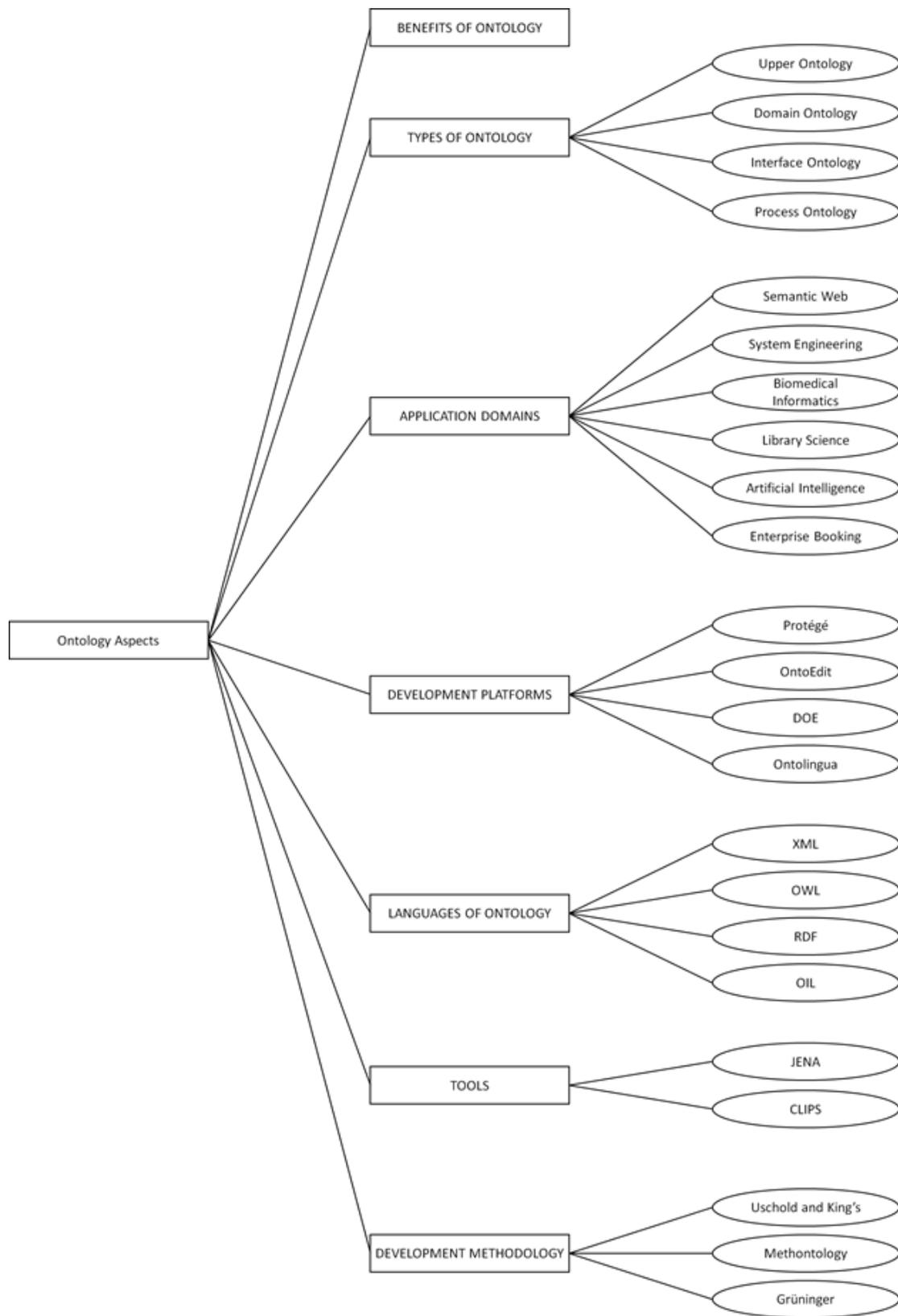


Fig. 1. The Taxonomy of Literature on Ontology Development Aspects.

## II. BENEFITS OF ONTOLOGY

One of the benefits is during the development of terminologies resources. Ontology could improve the content of built terminologies including its resources. According to Leonardi [7], the ambiguity of it able to be attributed to interdisciplinary and considered as most relevant way in expressing valuable resources in terminology aspect.

Author in [8] mentioned that ontology can assist in clarifying any structure of knowledge. The knowledge could be shared by using or through ontology that has been developed. To ensure the benefit is concrete, the author had done analysis by one satisfactory set of conceptualizations which includes respective terms from certain area of knowledge such as electronic-devices domain. In result, the ontology had captured specific knowledge bases by describing certain situation like various electronic devices manufacturers uses common vocabulary and syntax which product description's catalog was build.

According to [9], the features of ontology that is known to have essential relationship between concepts built into them are considered as its benefits. The feature enables the ontology to automate reasoning about data. In addition, it is easier to implement into semantic graph databases. The website also mentioned that ontology supports a variety of data representations such as unstructured, semi-structured or structured data which include better data integration, text-mining and data-driven analytics.

## III. TYPES OF ONTOLOGY

There are four types of ontologies, namely, Upper Ontology, Domain Ontology, Interface Ontology and Process Ontology [10].

Upper Ontology is an ontology defined as general concept to serve as common ground for communication specifically across different domains [12]. This raises questions of existing upper ontologies whether it could deal either as in part or as whole with social concepts. For example, "follower" in micro-blogging platform Twitter. Moreover, it consists of features like collaboration, federation and organization in a way to express the social policies and structures that could only be understood by others. [11] Nevertheless, upper ontology consists various types that are known as Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE), Basic Formal Ontology (BFO), General Formal Ontology (GFO), Yet Another More Advanced Top-level Ontology (YOMATO), Unified Foundational Ontology (UFO), PROTo Ontology (PROTON), Suggested Upper Merged Ontology (SUMO), Cyc and WordNet [13].

The second ontology is Domain Ontology. Unlike upper ontology, this ontology is a concept that belongs to a certain part of the world such as politics. Each domain ontology is modeled with specific definitions of terms [13]. Besides that, domain ontology consists of a collection of vocabularies and specifications of conceptualization of a given domain [14]. According to an article [15], the concepts and relationship between concepts within certain subject can be specified. The researcher emphasized that the model should closely be

matched with the level of information found in a textbook of that subject.

Process Ontology is understood as a description of components and relationship that able to make up a process [13]. Moreover, this ontology known as specification of entities and its relationship that represents information including process and formal representation of domain processes. By this, it is able to incorporate a comprehensive and formal knowledge of an organization's processes, activities, roles, application systems and much more. Nevertheless, it requires the hand of time and effort to extract information like roles, documents and so on from unstructured text that contains any process information [17].

The last type of ontologies is known as Interface Ontology. It is an explicit statement of patterns that provide valuable opportunity on reflecting current methods. Besides, it utilizes the basic categories of physical existence that could explain any less understandable and experiences to ensure the ontology is able to do it. Physical metaphors are presented in a pattern form [18].

## IV. APPLICATION DOMAINS

Ontology consists of multiple types. Widely known is Semantic Web. Based on a research done by M.M. Teye [19], Semantic Web is a heterogeneous and distributed that brought the evolution of web into high level. Whereby the ontology plays an important role in semantic web. Another research [20] mentioned that semantic web is to enhance human and machine interaction by representing data in understandable way for machine to mediate data and services. Nonetheless, without the help of ontology, domain schemas could not be represented formally.

Another application is ontology in agent communication language (ACL) [51] [52] [53] [54] [55] [56]. Ontology is widely used in multi-agent communication protocol [57] [58] [59] [60]. It provides a concise description of exchanged messages between different agents [61] [62] [63] [64] [65].

Apart from that, System Engineering is an engineering discipline that responsible for creating and executing interdisciplinary processes to ensure stakeholders and customers are satisfied with the system throughout the system's life cycle [21]. Additionally, the role of ontology in system engineering is to become an intermediary between organizations and people by assisting them in the process of maintaining and building systems that could become as an inter-operability among systems. [22].

Another ontology application is Biomedical Informatics. It provides support of common vocabulary of biomedical concepts, definitions, relationships, axioms and rules that could control the knowledge flow into the knowledge base. [23]. Researcher Rishi Kanth Saripalle said that the ontology is used to partially solve data management issue in medical. Prior to that, GALEN ontology provides terminology for building blocks of terminology description also combination of concepts description. [23].

Ontology also plays a role in Library Science. In this regard, it supports easy accessibility and re-use of existing

ontologies [24]. Similarly with others, the ontology provides a formal conceptualization of data that able to be shared, aligned and reused in this application. As a result, a well-established and number of ontology development increases [25].

In Artificial Intelligence applications, the ontology helps to eliminate problems that are semantic and machine-understandable representation of knowledge. The ontology facilitates knowledge by reusing and sharing knowledge, thus becoming a potential solution. With that, it acts as a link between information and users through logical abstracting information. [26].

Finally, in Enterprise Booking ontology, according to [27], the ontology presents a dynamic engineering design process while it constantly evolves, a collaborative ontology engineering tools is expected.

### V. DEVELOPMENT PLATFORMS

There are many platforms to support ontology design and development. One of those is Protégé. Protégé is an open source with free tool that supports huge community of active users. [29] Fig. 2 shows the interface of Protégé. Protégé provides an intuitive editor for ontology development with the help of tool extensions for ontology visualization, software engineering and other modelling tasks [30]. In addition, it provides a comprehensive Java-API to ensure it works with OWL and RDF models [31].

Meanwhile OntoEdit enables browsing, creating, maintaining and managing ontologies. However, a plugin architecture is required to be installed to ensure it is extensible and flexible for its purpose [29] With the plugin installed, OntoEdit not only ensure flexibility and extensibility but also able to export or import any ontologies that are in different formats such as RDF(s), XML and many more. The interface of OntoEdit is very similar to protégé, as shown in Fig. 3.

Although the interface layout is similar but the tools that are provided is different. While Protégé consists of “SQL Query” tab, “OntoGraf”, “Classes” tab and so on, OntoEdit offers tabs that are nearly the same as Microsoft Words like fonts, size of fonts, color of fonts and much more [29].

Next is Differential Ontology Editor (DOE). The main advantage of this platform is that it allows users to attach a lexical definition to relations, concepts, and justify the hierarchy. Despite having close name to OntoEdit, the interface of this platform is different as presented in Fig. 4.

The platform provides an import functionality. The ontology could be in form of XSLT, RDF(s), OIL, DAML+OIL and much more. This goes the same as the export function. DOE offers other features such as detection of cycles in hierarchices and ontology loading via URL [34].

Finally, Ontolingua is known to have a purpose of writing ontologies in canonical format. This is because it can be turned into a reasoning system with variety of representation. In addition to that, it allows one to maintain ontology into a machine-readable form while using different syntax and reasoning capabilities. Ontolingua is a public domain tool to define any common ontology. The syntax of ontolingua is

based on standard notation and semantics for predicate calculus. With that, the platform is able to translate and implement into a representation systems and also captures common representational conventions such as inheritance hierarchies and inverse relation maintenance. Ontolingua enable renaming of non-logical symbols from multiple components of ontology. Thus, this platform supports cyclic graphs and extends ontologies in many ways [35, 36].

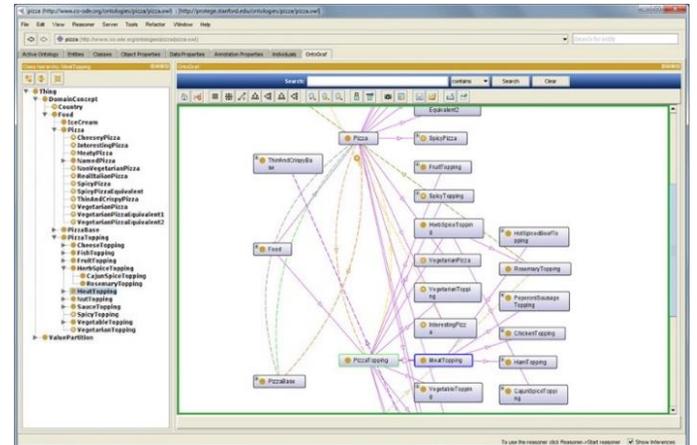


Fig. 2. Interface of Protégé [28].

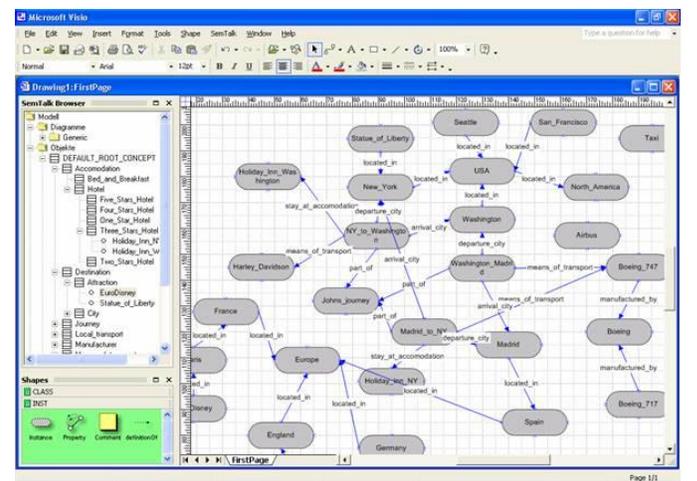


Fig. 3. Interface of OntoEdit[32].

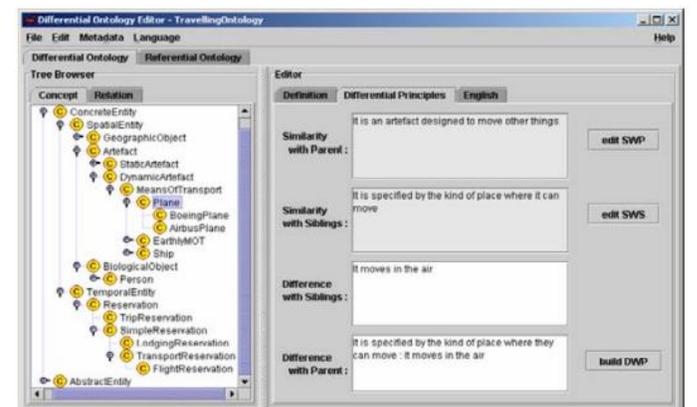


Fig. 4. Interface of DOE [33].

## VI. LANGUAGES OF ONTOLOGY

XML language is well-known. This language is a meta-language that facilitates the development of specialized tag languages. XML is strongly used in web specially to exchange data between applications [37]. Based on a study [37], the language was purposely designed to markup documents with arbitrary structures in opposed to HTML. The researcher also highlighted that this language creates a balance tree of nested sets of open and close tags. With that, each of the attribute value pairs without fixed vocabulary or set of permitted combinations can easily be defined. The syntax of this language is understandable by both platforms and human. Fig. 5 presents the syntax of the XML language.

The purpose of syntax is used to access the structure of documents. Any relevant information such as path expressions or templates for navigating are easier to locate within the document tree [38].

Next is Web Ontology Language (OWL). As mentioned by Techpedia, OWL was built upon standard with World Wide Web Consortium called Resource Description Framework (RDF) and have been evolved to many syntaxes and specifications. OWL has attracted great deals of interests from different fields such as medicine and academia [39]. This language was built on top of RDF and written XML. Based on a study, it is considered as part of the semantic web vision with the designed to be interpreted by computers. In result of the development, the purpose of OWL is to instantiate and define web ontologies with the description of properties, classes and instances of ontology [39] Although OWL is related to RDF and XML, the syntax or axioms however is entirely different than those two languages. Fig. 6 shows a sample of the OWL axioms.

```
WHERE
  <people>
    <person>
      <name>$P</name>
      <know-how>SGML</know-how>
    </person>
  </people> IN "some URL"
CONSTRUCT $P
```

Fig. 5. Illustration of XML Language Syntax

```
< owl : Class   rdf : about = "Author" / >
< owl : Class   rdf : about = "Paper" / >
< owl : ObjectProperty   rdf : about = "hasAuthor" / >
< owl : DatatypeProperty   rdf : about = "hasName" / >
< Author   rdf : about = "Bob" / >
< rdf : Description   rdf : about = "Bob" >
  < rdf : type   rdf : resource = "Author" / >
< /rdf : Description >
```

Fig. 6. Axioms of OWL Language.

The basic OWL construction is axioms. Based on Fig. 5, the author explained that Axioms (1) and (2) is the Author and Paper concepts that are presented in OWL Class. Meanwhile, Axioms (3) and (4) stated that hasAuthor is an object property and hasName is a data property. While, Axioms (5) and (6) are two different methods to assert the individual Bob that is an instance of class Author [40].

Another popular language is Resource Description Framework Shema (RDF). According to the study [41], RDF offers a distinguished vocabulary to model class and property hierarchies and other basic schemas from RDF models. This language also defines and permits modelling of objects with clear definition of semantics. RDF is usually accompanied by RDFS language. Based on a study [42], RDFS is known as an extension for RDF that consists of ability to give vocabularies, taxonomies and ontologies. Although both were written in XML syntax and used in Protégé tool, there are few differences between these two languages. Table I shows the comparison of RDFS and RDF.

The similarities of these languages are class, data, objects, domain and range, annotation and individual properties. While the contrast of the languages is the inverse properties, RDF does not contain it but RDFS does. Moreover, RDFS is better in inferencing, indexing and searching. Finally, SPARQL query and DL query, RDF does not consist any of those two queries while RDFS has them [41].

Next is OIL language. This language is the production of OnTo-Knowledge plan. It can unify three crucial aspects. These aspects are Description Logic, frame-based system supplied modelling language and web standard based on syntax of XML and RDF(s). [42]. Prior to that, OIL was specially designed to present the exact machine-accessible semantics of information of the web [43]. The architecture of OIL language makes it unique compared to others. There are three architectures. First is Standard OIL, whereby it is aimed to capture important mainstream modelling primitives to ensure adequate expressive power. Second, is Instance OIL. This layer includes individual integration with a modelling that could construct and specify individual fillers in terms of definition. Last architecture, Heavy OIL includes additional representational and reasoning capabilities [44]. Fig. 7 shows a sample code of OIL language [45].

TABLE I. COMPARISON BETWEEN RDF AND RDFS

Properties	RDFS	RDF
Class	Yes	Yes
Data	Yes	Yes
Object	Yes	Yes
Domain And Range	Yes	Yes
Annotation	Yes	No
Individual	Yes	Yes
Graph	RDFS	RDF
Inverse	Yes	No
Inference	Good	Poor
Indexing	Good	Poor
Searching	Good	Poor
SPARQL Query	Yes	No
DL Query	Yes	No
Prefix	RDFS	RDF

```

<rdfs:Class rdf:ID="herbivore">
  <rdf:type
    rdf:resource="http://www.
    ontoknowledge.org/oil/RDFS-
    schema/#DefinedClass" />
  <rdfs:subClassOf rdf:resource="#animal" />
  <rdfs:subClassOf>
    <oil:NOT>
      <oil:hasOperand rdf:resource="
        #carnivore" />
    </oil:NOT>
  </rdfs:subClassOf>
</rdfs:Class>

```

Fig. 7. Sample Code of OIL.

To elaborate, the OIL expression defines ‘Herbivore’ as a class and a subclass for ‘Animal’ and a disjunct to ‘Carnivores’.

### VII. DEVELOPMENT TOOLS FOR SEMANTIC WEB

There are two well-known platforms for building Semantic Web. The first platform is JENA. It is a Java-based language platform with the purpose of creating applications for semantic web. This platform provides interfaces and classes for the manipulation and creation of RDF repositories. It also includes the capability of creating and manipulating RDF graphs. Furthermore, the semantic data can interact with the libraries contain inside JENA [46].

According to a study [47], JENA supports many description languages like DAML+OIL, RDFS, OWL and much more. Fig. 8 shows some part of the JENA codes that is used to create ontology.

```

GetStartedSemWeb.java
19 import java.io.IOException;
2 public class GetStartedSemWeb {
3     static String defaultNamespace = "http://semanticweb.org/ontologies";
10 Model schema = null;
11     public static void main(String[] args) throws IOException
12     {
13         GetStartedSemWeb getsemweb = new GetStartedSemWeb();
14         System.out.println(" Adding student ontology ");
15         getsemweb.loadontology();
16     }
17     private void loadontology() throws IOException{
18         schema = ModelFactory.createOntologyModel();
19         java.io.InputStream inschema = FileManager.get().open("C:/Users/AYESHA ANEEN/Desktop/docur
20         schema.read(inschema, defaultNamespace);
21         ExtendedIterator it = ((OntModel) schema).listClasses();
22         while(it.hasNext())
23         {
24             OntClass cls= (OntClass)it.next();
25             System.out.println("URI of classes of Student Ontology are "+cls.getURI());
26

```

Fig. 8. Sample Code of JENA to Create Ontology.

```

(defclass <name>
  (is-a <superclass-name>+)
  (<slot>* <constraint>*)
  (<multislot>* <constraint>*))

```

Fig. 9. Basic Syntax for Defining Classes in COOL.

In Fig. 8, the codes consist of all necessary import to supporting Semantic Web applications. GetStartedSemWeb class is created for next default namespace and model. This class is to hold semantic data that have been declared while getsemweb in the main method. It is an object created and call made to loadontology method. Meanwhile, loadontology is another ontology created by calling ModelFactory. Create OntologyModel(). Aside from that they will call up another method to show the result of the ontology. JENA is not the only platform used for building Semantic Web, there is another one named ‘CLIPS’. In an article explained that CLIPS is a RETE-based production rule engine that was written in C language. One interesting fact of this tool is that it able to integrate the production rule paradigm with OO model that can be defined by using COOL language (CLIPS Object-Oriented Language). CLIPS provide the necessary constructs to define classes with attributes and corresponding data model.

Fig. 9 shows the basic syntax of defining classes using COOL language. The attributes in COOL can take either a single value called slots or more than one value called multi slots. COOL regarded classes as type of objects and may also define it as attributes. Whereby, the classes of domain are usually organized in terms of class hierarchy by using subclass relationships that may have more than one direct super classes.

This tool is suitable for applications whereby ontologies are used as static models for sharing knowledge in heterogeneous environments. For example, the domain of Software Antipatterns [48].

### VIII. ONTOLOGY DEVELOPMENT METHODOLOGY

This section discusses three popular methodologies for ontology development. First is Uschold and King’s methodology. This methodology has been developed based on Enterprise Ontology experience. Hence, able to provide guidelines for developing ontology [16]. Nevertheless, this methodology provides three phases for users to easily understand and follow the flow of a development process. Table II shows the summary of the three phases.

Meanwhile for Methontology, it was developed by an Ontological Engineering group at Universidad Politécnic de Madrid. Different with the previous methodology, Methontology enables the construction of ontologies at knowledge level which includes the root of the main activities that was identified by IEEE software development process and other knowledge engineering methodologies. Nevertheless, this methodology consists of six more steps than Uschold and King’s methodology. All the steps are presented in Table III.

Methontology could identify management activities such as schedule, quality assurance, control and support activities [49]. Finally, this methodology also uses user-friendly approach in acquisition knowledge by non-knowledge engineer. This shows that the approach is an effective yet applicable method especially for domain-knowledge-model construction [50].

TABLE. II. SUMMARY OF THE PHASES USED IN USCHOLD AND KING METHODOLOGY

Phase	Summary
Identifying purpose	To clarify the reason why the ontology is built and what is the purpose of use of the ontology.
Ontology Capture	<ul style="list-style-type: none"><li>To identify all the key concepts and relationships between a domain.</li><li>Precise unambiguous text definitions and identifying terms to refer to concepts and relationships in first phase.</li></ul>
Coding	Activities involved in explicitly representing knowledge
Integrating Existing Ontologies	Capture and coding process are involved and two questions are highlighted that is “how” and “whether” to use existing ontology
Evaluation	<ul style="list-style-type: none"><li>Adopts the definition of making technical judgement of the ontologies and associate it with software environment and documentation</li></ul>
Documentation	All the guidelines of the ontology are established

TABLE. III. SUMMARY OF GRÜNINGER AND FOX METHODOLOGY

Steps	Label	Summary
1.	Capture of motivating scenarios	Development is mostly triggered by scenarios that tend to rise the application. Moreover, the scenarios are considered as examples of problems that didn't addressed adequately.
2.	Formulation of informal competency questions	All form of questions is based on previous scenarios and considered as expressiveness requirements. Most of the questions are stratified and can be used to answer general questions from another ontology.
3.	Specification of the terminology of the ontology	All informal terminologies are obtained. Whereby set of terms are extracted from the questions.
4.	Formulation of formal competency questions by using terminology	The competency of questions and the terminology of the ontology are formally defined
5.	Specification of definitions and axioms	Axioms and definition are arranged into a first-order sentences and are used to define the constraints and terms for the objects inside the ontology.
6.	Establish conditions for characterizing the completeness of the ontology	All questions and conditions have been formally stated and defined.

The third methodology is by Grüninger and Fox [16]. This methodology was created according to an experience of developing the TOVE project ontology. There are six steps in this methodology, as summarized in Table III.

#### IX. DISCUSSION AND RECOMMENDATION

Based on review, each of the ontology types has their own purpose. For example, domain ontology can be used to describe concepts, relationship, definitions with specific definition of a term. With that, domain ontology could assist any field such as Artificial Intelligence, Library Science, Biomedical Informatics and much more as these sectors are dealing a lot with definitions, terms and so on. However, for Upper Ontology, this particular serves as common ground for communication across different domains such as twitter or any organization that is required to communicate from different domains. Meanwhile, for process ontology is used to express the components and relationship that can make up as a process. With its capability to represent information of process and domain processes, an organization can apply this ontology to describe their role, processes, activities and much more of the organization. Nevertheless, to reflect any explicit patterns that could possibility provide valuable opportunity on current methods, it is best to use Interface Ontology as it uses basic categories of physical existence.

In order to develop the ontology, it is required to have a platform for development. As mentioned in this paper, there many tools that can be used to develop an ontology, though it depends on the developer's preference and needs. One of the

most known platform among developers is protégé. As it is a free tool software included with tools extension to assist in visualization, software engineering and other modelling tasks. However, if a developer is requiring to manage or maintain an ontology, OntoEdit is the suitable platform to do so as it has the capability to manage and maintain but a plugin architecture is required to ensure the platform is flexible and extensible for the purpose. Another platform is Differential Ontology Editor (DOE), this tool can assist developers to create a lexical definition to relations, concepts, and justify the hierarchy. In addition to that, if developers need to import any ontology in any format, this tool can provide an import function for the purpose. Next, for ontology that is need to be written in a canonical format, Ontolingua is the tool to use as it allows maintain an ontology into a machine-readable form by using reasoning capabilities and different syntaxes.

In connecting with the developing platform, there are also various languages that can be used by the developers, according to their preference. To facilitate data exchange between applications, common language to use is XML language. This language specialized in tag languages and also creates a balance tree of nested sets of close and open tags. With that users are not bound to define every attribute pairs with fixed vocabularies. Furthermore, another language that can be use is Web Ontology Language (OWL) with the combination of RDF and XML, users able to define and instantiate web ontologies by describing the instances, properties and classes of a particular ontology. Now, given that OWL is combination of RDF and XML, RDF can also be used as stand-alone. For RDF, the language able to define and allow

REFERENCES

modelling objects with clear definition of semantics. Normally, when developers used this language, they will be accompanied by an extension that provide an ability to give taxonomies, ontologies and vocabularies called RDFS. Besides, this language can be used in Protégé software. If users are dealing with machine-accessible semantics information of the web, the language that has the functionality to do so is OIL language. Unlike the other languages, OIL consist three architecture known as Standard OIL, Instance OIL and Heavy OIL. Each of the architecture has their own role in this language to carry out the task.

In any case if the developer is interested to build an ontology in semantic web, there are two softwares can be used. One is a Java-based platform known as JENA that can handle multiple description language (ie: RDFS, OWL etc). Second is CLIPS, a C-language tool that able to integrate a production rule paradigm with COOL language suitable for static model ontology in sharing knowledge in a heterogeneous environment. As we known that ontology is a development process.

Therefore, to assist developers in building ontology, there are three methodologies to use. The common methodology is Uschold and King's methodology that provide a guideline of each phases in developing ontology. In comparison with Methontology, this method able to identify management activities and it is said to be an effective yet applicable approach for domain-knowledge model construction. Lastly, similar to both methodologies, the Grüniger and Fox Methodology was created based on the experience of TOVE project ontology that has six steps, begin with capturing motives and ending with established condition. To conclude, ontology can be developed through various ways according to the suitability and preference of a certain project and developers. Unlike any other development projects, ontology is dynamic and flexible that is not bound to a certain development process or progress to be created.

X. CONCLUSION

An overview of the ontology development aspects has been presented in this paper. The definition of ontology was discussed from different perspectives. Additionally, various types of ontology are available to be known as Upper Ontology, Domain Ontology, Interface Ontology and Process Ontology was mentioned. Moreover, all types of ontology discussed have applied in various domain such as Semantic Web, System Engineering and Biomedical Informatics. Furthermore, three common development platform and designing ontology which are Protégé, OntoEdit, Differential Ontology Editor (DOE) and Ontolingua are also presented in this paper. Besides that, basic languages are also discussed. Finally, from various development tools to languages, both are used together to build an ontology by several methods which are Methontology, Uschold and King's and Gruniger and Fox.

ACKNOWLEDGMENTS

This work is sponsored by Tenaga Nasional Berhad (TNB) under TNB R&D Seeding Fund Scheme No. TC-RD-18-19. We also gratefully appreciate Universiti Tenaga Nasional & Uniten R&D for securing and managing the fund.

- [1] "Birth of a New Science: The History of Ontology from Suárez to Kant," *Ontology: Theory and History*. [Online]. Available: <https://www.ontology.co/history.htm>. [Accessed: 07-Jul-2018].
- [2] J. Busse et al., "Actually, what does 'ontology' mean?: A term coined by philosophy in the light of different scientific disciplines," *J. Comput. Inf. Technol.*, 2015.
- [3] N. Guarino, D. Oberle, and S. Staab, "Handbook on Ontologies," no. May 2009, pp. 0–17, 2009.
- [4] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, 1993.
- [5] D. Man, "Ontologies in Computer Science," *Didactica Mathematica*, vol. 31, no. 1, pp. 43–46.
- [6] B. Smith, "Ontology," *Blackwell Guide to the Philosophy of Computing and Information*, pp. 1–13, 2003.
- [7] I. D. Muñoz and M. R. B. Zambrana, "Applying Ontologies to Terminology: Advantages and Disadvantages," *Hermes – Journal of Language and Communication in Business*, pp. 66–76, Dec. 2013.
- [8] B. Chandrasekaran, J. R. Josephson and V. R. Benjamins, "What are ontologies, and why do we need them?" in *IEEE Intelligent Systems and their Applications*, vol. 14, no. 1, pp. 20–26, Jan.-Feb. 1999.
- [9] "What are Ontologies and What are the Benefits of Using Ontologies," *Ontotext*. [Online]. Available: <https://ontotext.com/knowledgehub/fundamentals/what-are-ontologies/>. [Accessed: 07-Jul-2018].
- [10] Beisswanger, E., Schulz, S., Stenzhorn, H., & Hahn, U. (2008). *BioTop: An upper domain ontology for the life sciences*. *Applied Ontology*, 3(4), 205–212.
- [11] J. Grabarske and D. Heutelbeck, "An upper ontology for the social web," in *Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2012*, 2012.
- [12] Mascardi, V., Cordì, V., & Rosso, P. (2007, September). *A Comparison of Upper Ontologies*. In *Woa (Vol. 2007)*, pp. 55–64.
- [13] Fensel, D. (2001). *Ontologies*. In *Ontologies* (pp. 11–18). Springer, Berlin, Heidelberg.
- [14] "What is Domain Ontology," IGI Global. [Online]. Available: <https://www.igi-global.com/dictionary/ontology-based-multimodal-language-learning/8239>. [Accessed: 01-Aug-2018].
- [15] S. Boyce and C. Pahl, "Developing Domain Ontologies for Course Content," *Developing Domain Ontologies for Course Content. Educational Technology & Society*, pp. 275–288, 2007.
- [16] M. Fernández López, "Overview Of Methodologies For Building Ontologies," pp. 1–13.
- [17] [O. Gurbuz and O. Demirors, "A comparison of process ontology discovery from organizational guidelines in two different languages," in *Proceedings - 43rd Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2017*, 2017.
- [18] P. Barr, R. Biddle, and J. Noble, "Interface Ontology: Creating a Physical World for Computer Interfaces."
- [19] M. M. Taye, "Understanding Semantic Web and Ontologies: Theory and Applications," vol. 2, no. 6, pp. 182–192, 2010.
- [20] Y. Ding, "Semantic Web: Who is who in the field – A bibliometric analysis," *e Journal of Information Science*, pp. 1–22, 2009.
- [21] What is Systems Engineering? <http://www.incose.org/AboutSE/WhatIsSE>
- [22] O. Chourabi, Y. Pollet, and M. Ben Ahmed, "Ontology based knowledge modeling for System Engineering projects," *2008 Second Int. Conf. Res. Challenges Inf. Sci.*, pp. 453–458, 2008.
- [23] R. Saripalle, "Current status of ontologies in Biomedical and Clinical Informatics," *Engr.Uconn. Edu*, vol. 16, pp. 1–14, 2013.
- [24] Y. Ding and D. Fensel, "Ontology Library Systems: The key to successful Ontology Re-use," *Proc. SWWS*, pp. 93–112, 2001.
- [25] M. d'Aquin and N. F. Noy, "Where to publish and find ontologies? A survey of ontology libraries," *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 11, pp. 96–111, Mar. 2012.
- [26] Y. Ding, "IR and AI: The role of ontology," pp. 1–11.

- [27] B. Ell, E. Simperl, S. Wolger, B. Kampgen, S. Hangl, D. Vrandečić, and K. Siorpaes, "Context and Semantics for Knowledge Management," *Context and Semantics for Knowledge Management*, pp. 29–59.
- [28] <https://protege.stanford.edu/assets/img/screenshots/desktopprotege-screenshot-5.jpg>
- [29] A. L. N. Escórcio and J. Cardoso, "Editing Tools for Ontology Creation," *Semant. Web Serv. Theory, Tools Appl.*, no. March, pp. 71–95, 2007.
- [30] R. Sivakumar and P. V. Arivoli, "ONTOLOGY VISUALIZATION PROTÉGÉ TOOLS-A REVIEW," *Int. J. Adv. Inf. Technol.*, vol. 1, no. 4, 2011.
- [31] M. A. Musen and R. D. Stevens, "The Protege OWL Experience," *Proc. OWLED. Work. OWL Exp. Dir.*, no. January, 2005.
- [32] <http://www.semtalk.com/pub/EON2003experiment-Dateien/image012.jpg>
- [33] S. Youn, A. Arora, P. Chandrasekhar, P. Jayanty, A. Mestry, and S. Sethi, "Survey about Ontology Development Tools for Ontology-based Knowledge Management."
- [34] S. Youn and D. Mcleod, "Ontology Development Tools for Ontology-Based Knowledge Management," *Encycl. E-Commerce*, 2006.
- [35] A. Farquhar, R. Fikes, and J. Rice, "Tools for Assembling Modular Ontologies in Ontolingua."
- [36] D. Fensel, F. van Harmelen, I. Horrocks, D. L. McGuinness, and P. F. Patel-Schneider, "OIL: An Ontology Infrastructure for the Semantic Web."
- [37] Decker et al., "Semantic Web: The roles of XML and RDF," *IEEE Internet Comput.*, vol. 4, no. 5, pp. 63–74, 2000.
- [38] M. Erdmann and R. Studer, "How to structure and access XML documents with ontologies," *Data Knowl. Eng.*, 2001.
- [39] "What is Web Ontology Language (OWL)?-Definition from Techopedia," *Techopedia.com*. [Online]. Available: <https://www.techopedia.com/definition/843/web-ontology-language-owl>. [Accessed: 07-Jul-2018].
- [40] K. Sengupta and P. Hitzler, "Web Ontology Language (OWL). Encyclopedia of Social Network Analysis and Mining," 2014.
- [41] M. M. Aref and Z. Zhou, "The Ontology Web Language (OWL) for a multi-agent understating system," in *2005 International Conference on Integration of Knowledge Intensive Multi-Agent Systems, KIMAS'05: Modeling, Exploration, and Engineering*, 2005.
- [42] S. Staab, M. Erdmann, A. Maedche, and S. Decker, "Ontologies in rdf (s)," *ETAI J.*, vol. 6, pp. 1–17, 2001.
- [43] [W. Bin, "The discussion on transforming from OIL ontology to basic-element and complex-element of Extenics," in *ICSESS 2012 - Proceedings of 2012 IEEE 3rd International Conference on Software Engineering and Service Science*, 2012.
- [44] J. A. Khan and S. Kumar, "Deep analysis for development of RDF, RDFS and OWL ontologies with protege," *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization*, Noida, 2014, pp. 1-6.
- [45] O. Lassila, F. van Harmelen, I. Horrocks, J. Hender, and D. L. McGuinness, "The semantic Web and its languages," *IEEE Intell. Syst.*, vol. 15, no. 6, pp. 67–73, 2000.
- [46] D. Fensel, F. van Harmelen, I. Horrocks, D. L. McGuinness, and P. F. Patel-Schneider, "OIL: An Ontology Infrastructure for the Semantic Web."
- [47] A. Ameen, K. U. R. Khan, and B. P. Rani, "Extracting knowledge from ontology using Jena for semantic web," in *2014 International Conference for Convergence of Technology, I2CT 2014*, 2014.
- [48] S. Zhang, J. Guo, Z. Yu, C. Lei, C. Mao and H. Wang, "An Approach of Domain Ontology Construction Based on Resource Model and Jena," *2010 Third International Symposium on Information Processing*, Qingdao, 2010, pp. 311-315.
- [49] O. Corcho, M. Fernández-López, A. Gómez-Pérez, and A. López-Cima, "Building legal ontologies with METHONTOLOGY and WebODE," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2005.
- [50] M. F. López, A. Gómez-Pérez, J. P. Sierra, and A. P. Sierra, "Building a chemical ontology using methontology and the ontology design environment," *IEEE Intell. Syst. Their Appl.*, 1999.
- [51] Mahmoud MA., Ramli R., Azman F, and Grace J. (2018). A Development Methodology Framework of Smart Manufacturing Systems (Industry 4.0), MySEC 2018.
- [52] Ahmad A, Zaliman M, Yusof M, Ahmad MS, Ahmed M, Mustapha A. Resolving conflicts between personal and normative goals in normative agent systems. In2011 7th International Conference on Information Technology in Asia 2011 Jul 12 (pp. 1-6). IEEE.
- [53] Ahmed M, Ahmad MS, Yusoff MZ. Modeling agent-based collaborative process. InInternational Conference on Computational Collective Intelligence 2010 Nov 10 (pp. 296-305). Springer, Berlin, Heidelberg.
- [54] Jassim OA, Mahmoud MA, Ahmad MS. A multi-agent framework for research supervision management. InDistributed Computing and Artificial Intelligence, 12th International Conference 2015 (pp. 129-136). Springer, Cham.
- [55] Mahmoud MA, Ahmad MS, Ahmad A, Yusoff MZ, Mustapha A. A norms mining approach to norms detection in multi-agent systems. In2012 International Conference on Computer & Information Science (ICCIS) 2012 Jun 12 (Vol. 1, pp. 458-463). IEEE.
- [56] Ahmed M, Ahmad MS, Yusoff MZ. Mitigating human-human collaboration problems using software agents. InKES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications 2010 Jun 23 (pp. 203-212). Springer, Berlin, Heidelberg.
- [57] Mahmoud MA, Ahmad MS, Yusoff MZ. A norm assimilation approach for multi-agent systems in heterogeneous communities. InAsian Conference on Intelligent Information and Database Systems 2016 Mar 14 (pp. 354-363). Springer, Berlin, Heidelberg.
- [58] Mahmoud MA, Ahmad MS, Yusoff MZ, Idrus A. Automated multi-agent negotiation framework for the construction domain. InDistributed Computing and Artificial Intelligence, 12th International Conference 2015 (pp. 203-210). Springer, Cham.
- [59] Mahmoud MA, Mustapha A, Ahmad MS, Ahmad A, Yusoff MZ, Hamid NH. Potential norms detection in social agent societies. InDistributed Computing and Artificial Intelligence 2013 (pp. 419-428). Springer, Cham.
- [60] Mahmoud M, Ahmad MS, Yusoff MZ. Development and implementation of a technique for norms-adaptable agents in open multi-agent communities. *Journal of Systems Science and Complexity*. 2016 Dec 1;29(6):1519-37.
- [61] Mostafa SA, Ahmad MS, Annamalai M, Ahmad A, Gunasekaran SS. A dynamically adjustable autonomic agent framework. InAdvances in Information Systems and Technologies 2013 (pp. 631-642). Springer, Berlin, Heidelberg.
- [62] Mostafa SA, Darman R, Khaleefah SH, Mustapha A, Abdullah N, Hafit H. A General Framework for Formulating Adjustable Autonomy of Multi-agent Systems by Fuzzy Logic. InKES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications 2018 Jun 20 (pp. 23-33). Springer, Cham.
- [63] Mostafa SA, Ahmad MS, Ahmad A, Annamalai M, Gunasekaran SS. A Flexible Human-Agent Interaction model for supervised autonomous systems. In2016 2nd International Symposium on Agent, Multi-Agent Systems and Robotics (ISAMSR) 2016 Aug 23 (pp. 106-111). IEEE.
- [64] Mostafa SA, Ahmad MS, Annamalai M, Ahmad A, Gunasekaran SS. A conceptual model of layered adjustable autonomy. InAdvances in Information Systems and Technologies 2013 (pp. 619-630). Springer, Berlin, Heidelberg.
- [65] Mostafa SA, Ahmad MS, Ahmad A, Annamalai M, Mustapha A. A dynamic measurement of agent autonomy in the layered adjustable autonomy model. InRecent Developments in Computational Collective Intelligence 2014 (pp. 25-35). Springer, Cham.