International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Ontology Life Cycle: A Survey on the Ontology and its Development Steps

Eiman Alsiddig Altayeb Ibrahim[1], Mohammed Awad Mohammed Ataelfadiel[2]

[1]Sudan University of Science and Technology, College of Computer Science and Information Technology, Khartoum – Sudan

[2]Computer Science dept. AL-Imam MuhommedIbn Saud Islamic University, AL-Ahsa, Kingdom of Saudi Arabia

Abstract: ***In the last decades, the turnout has become great for the use of ontology in various fields, such as web technologies, database integration, multi agent systems, natural language processing, etc... The main goal of this paper is to answer generic questions about ontologies, such as: What is the ontology? Which are the different Types of ontologies? What is the purpose of the use of ontologies in an application? Which methods can I use to build ontology? During the paper we discussed the definition of Ontology, types of ontologies (Formal and Informal ontology) and its Component.Also, highlighted some of the languages used in the construction of the Ontologies beside the ontology Building Tools. In addition, we listed the agreed steps for building the Ontology. In conclusion, we did not overlook to mention some of the Ontology projects specifically in Holy Quran Domain.***

Keywords: Ontology Definition, Ontology Types, Ontology Building Tools, Ontology languages, Ontology components, Ontology Engineering

1. Introduction

Historically, ontologies arise out of the branch of philosophy known as metaphysics, which deals with the nature of reality – of what exists. This fundamental branch is concerned with analyzing various types or modes of existence, often with special attention to the relations between particulars and universals, between intrinsic and extrinsic properties, and between essence and existence. The traditional goal of ontological inquiry in particular is to divide the world "at its joints" to discover those fundamental categories or kinds into which the world's objects naturally fall [1].

Since the mid-1970s, researchers in the field of artificial intelligence (AI) have recognized that capturing knowledge is thekey to building large and powerful AI systems. AI researchers argued that they could create new ontologies as computational models that enable certain kinds of automated reasoning. In the 1980s, the AI community began to use the term ontology to refer to both a theory of a modeled world and a component of knowledge systems [1].

In the context of computer and information sciences, ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (orproperties), and relationships (or relations among class members). The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application. In the context of database systems, ontology can be viewed as a level of abstraction of data models, analogous to hierarchical and relational models, but intended for modeling knowledge about individuals, their attributes, and their relationships to other individuals [1].

2. Ontology Definition

There are many interpretations about what ontology is. In fact, hot discussions are often done in many meetings on ontology. However, the ontology community has come to an agreement on giving up its definition.

A compositional definition is given as follows:
An ontology consists of concepts, hierarchical (is-a) organization of them, relations among them (in addition to is-a and part-of), axioms to formalize the definitions and relations [2].

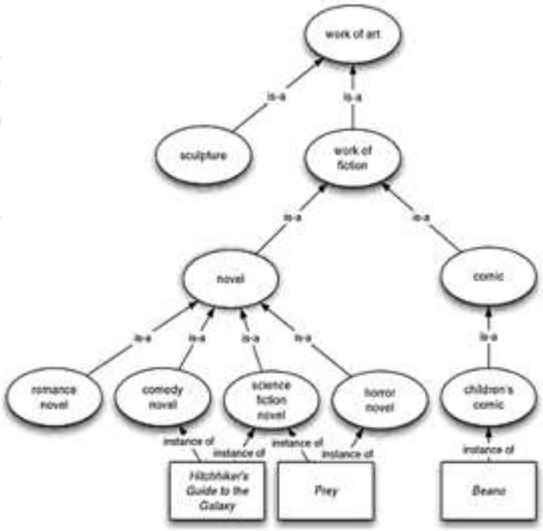


Figure1: classes are draw as oval, and instances as rectangles. Labels on arrowsindicate the nature of the relationship between entities [3].

3. Ontology of Ontologies

Depending on their intend use, ontologies come in variety of types, varying in their formality and their specificity.

3.1 Informal Ontologies

- **Controlled vocabulary** perhaps the simplest kind of ontology is controlled vocabulary. In a controlled vocabulary, we make use of a few predefined keywords to classify entities: no perhaps, properties, or axioms. In some domains, this is sufficient.
- **Terms/ glossary** here, a list of terms, as with a controlled vocabulary, but some attempt is made (typically with natural language, e.g. an English explanation) to define the meaning of these terms. However, the computational power of such ontology is roughly that of a controlled vocabulary, since we cannot in general compute with the natural language explanation. The value of the explanation is therefore usually for the ontology designer.
- **Thesaurus** a thesaurus defines synonyms: terms that have the same meaning. Thus, if you want to find a web service that provides weather forecasts, it might be useful to know that 'meteorological forecast' means the same thing.Some Thesaurus are: **URBAMET** Thesaurus, **GEMET** Thesaurus, **Agrovoc** Thesaurus [4]
- **Informal 'is-a' taxonomies** here we think of controlled vocabularies organized into an informal hierarchy. We find such hierarchies on web sites such as Amazon.com, for example, where goods for sale are organized into loose hierarchies. Typically the hierarchies are not formal hierarchies, because related goods (e.g. cameras and camera bag) are collected together in the same place, without any formal definition of how or why the goods are clustered in this way [3].

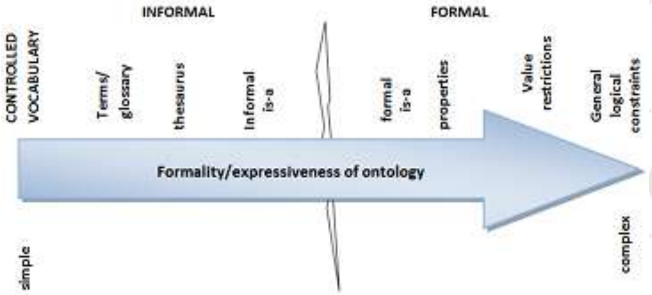


Figure 2: the ontology spectrum from informal up to formal expressive [3].

3.2 Formal Ontologies

Where some attempts is made to give the terms in the ontology some formal semantics.

- ***Formal 'is-a'taxonomies*** here, we explicitly define subsumption relationships between classes as shown in (figure 2).
- ***Properties*** here, we now allow classes to have properties, and together with the subsumption relation, this permits us to draw conclusions about the properties of classes.
- ***Value restrictions*** value restrictions give additional information about relationships; for example, a typical restriction might say that 'every person has exactly one birth mother'.
- ***Arbitrary logical constrains*** finally, we might have ontologies with arbitrary logical constrains. Such constrains go beyond value restriction, taxonomical hierarchies, etc. In general, such constrains allow us a great degree of precision when defining an ontology. The drawback with such constrains is that, in general allowing arbitrary logical expressions leads to very high computational complexity (and even undesirability) with respect to reasoning [3].

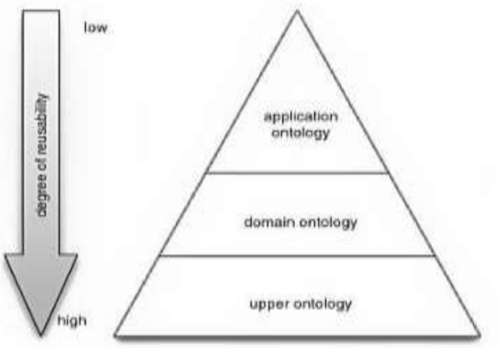


Figure 3: The ontology hierarchy: from (reusable) at the bottom, (not very reusable) at the top [3].

As well as distinguishing ontologies based on their formality and the types of information(Figure 2), can also usefully distinguish ontologies based on their role in an application (Figure 3). At the bottom, the most general kind of ontology: the so-called '*upperontology* ' (even though it appears at the bottom in (Figure 3) such an ontology might start out by defining the most general class and then define classes that specialize this.

Domain ontology defines concepts appropriate for a specific application domain, this research is domain ontology. For example, it might define concepts relating to Learning terminology, and be used by a number of applications in the area of Open Learning. Note that domain ontology will typically build upon and make use of concepts from an upper ontology: this idea of *reuse* of ontologies is very important, as the more applications use a particular ontology, the more agreement there will be on terms.

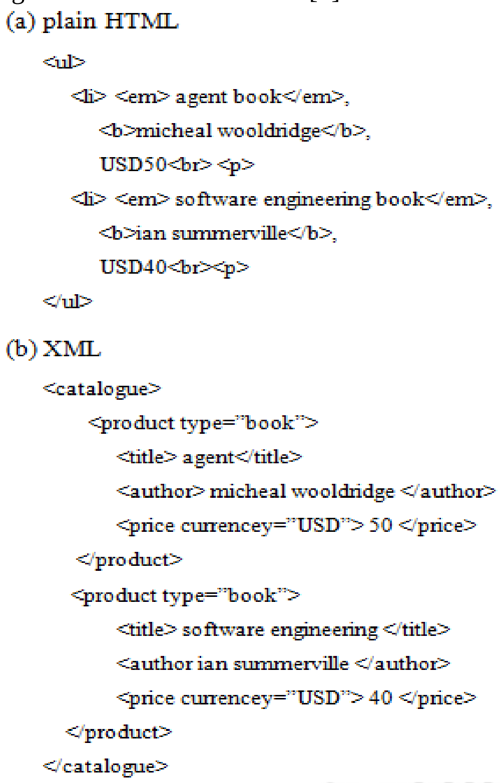
Finally, *application ontology* defines concepts used by a specific application. Again, it will typically build upon a domain ontology and it turn upon some upper ontology. Concepts from application ontology will not usually be reusable: they will typically be of relevance only within the application for which they were defined [3].

4. Ontology languages

4.1 XML – ad hoc ontologies

Arguably the simplest ontologies to create and use are *ad hoc* ontologies: create with little effort, for a specific purpose, usually with a short expected period of use. Often such ontologies take the form of a controlled vocabulary, and XML is usually the language of choice of such ontologies.

The extensible markup language (XML) is not an ontology language, although it can be directly used to define simple ontology in an informal way. It is best thought of as a kind of extension to HTML, which in a nutshell allows us to define our own tags and document structures [3].



```
(a) plain HTML
    <ul>
      <li> <em> agent book</em>,
        <b>micheal wooldridge</b>,
        USD50<br> <p>
      <li> <em> software engineering book</em>,
        <b>ian summerville</b>,
        USD40<br><p>
    </ul>
(b) XML
    <catalogue>
         <product type="book">
            <title> agent</title>
            <author> micheal wooldridge </author>
            <price currencey="USD"> 50 </price>
        </product>
       <product type="book">
            <title> software engineering </title>
            <author ian summerville </author>
            <price currencey="USD"> 40 </price>
       </product>
    </catalogue>
```

Figure 4: plain HTTL versus XML

It is clear that computer program would have a much easier time *understanding the meaning* of (Figure 4,b) than it would with (Figure 4,a), there is nothing to help a program understand which part of the document refers to the price of the product, which refers to the author of the product, and so on. We can think of the tags in (Figure 4, b) as being a controlled vocabulary, providing a useful but relatively informal ontology.

4.2 Ontolingua

Ontolingua is originally an Interlingua for ontology representation and sharing developed by KSL (Knowledge Systems Lab) at StanfordUniversity. It is designed by adding frame-like representation and translation functionalities to KIF (Knowledge Interchange Format)[KIF] which is a logic-based interlingua for knowledge representation. It can translate from and to some description logics languages such as Loom, Epikit, etc. Ontolingua itself does not have inference functionality. It has currently developed into a development environment which Provides a set of ontology development functions (browse, create, edit, modify and use ontologies) and a library of modular and reusable ontologies. Although it had been a key language for ontology representation for years since its development, it is not active recently because of the advent of XML family languages.

4.3 RDF(S)

Resource Description Framework (RDF) is a framework for metadata description developed by W3C (WWW Consortium). It employs the triplet model <object, attribute, value>, well-known in AI community, in which object is called resource representing a web page. A triplet itself can be an object and a value. Value can take a string or resource. Object and value are considered as a node and attribute as a link between nodes. Thus, an RDF model forms a semantic network. RDF has an XML-based syntax (called serialization) which makes it resembles a common XML-based markup language. But, RDF is different from such a language in that it is a data representation model rather than a language and that the XML's data model is the nesting structure of information and the frame-like model with slots.

Although RDF has been designed for metadata representation model, it can be used as a general-purpose knowledge representation, which might be apparent from the fact that it is a kind of semantic network model.

4.4 Web Ontology Language (OWL)

Web Ontology Language (OWL) is also a language developed by W3C. OWL is designed to make it a common language for ontology representation and is based on DAML+OIL. OWL is an extension of RDF Schema and also employs the triple model. Its design principle includes developing a standard language for ontology representation to enable semantic web, and hence extensibility, modifiability and interoperability are given the highest priority. At the same time, it tries to achieve a good trade-off between scalability and expressive power.
OWL is a collection of several XML-based ontology frameworks, within which ontologies in these various frameworks can be expressed. Specifically, there are three main level of OWL, as follows:

- **OWL Lite.** This is simplest (least expressive) variant of OWL, which supports only basic ontology features. In particular, OWL Lite places a number of restrictions on the type of axioms one can write. The point about these restrictions is that they result in a language that is computationally more tractable (and is also somewhat easier for human to use and understand) than more expressive OWL variants.
- **OWL DL.** This language extends the properties of OWL Lite. The features of OWL DL were carefully chosen so that

the language corresponds exactly to a particular formalism know as description logic.
- **OWL Full.** This is very expressive framework, providing many features for defining ontologies; however, in its full glory, the framework is so rich that many reasoning problems with OWL full (such as consistency checking) are understandable [3].

5. Ontology components

Contemporary ontologies share many structural similarities, regardless of the language in which they are expressed. As mentioned above, most ontologydescribes individuals (instances), classes (concepts), attributes, and relations. In this section each of these components is discussed in turn.

Common components of ontologies include:
- **Individuals:** instances or objects (the basic or "ground level" objects).
- **Classes:**sets, collections, concepts, classes in programming, types of objects, or kinds of things.
- **Attributes:** aspects, properties, features, characteristics, or parameters that objects (and classes) can have.
- **Relations:** ways in which classes and individuals can be related to one another.
- **Function terms:** complex structures formed from certain relations that can be used in place of an individual term in a statement.
- **Restrictions:** formally stated descriptions of what must be true in order for some assertion to be accepted as input.
- **Rules:** statements in the form of an if-then (antecedent-consequent) sentence that describe the logical inferences that can be drawn from an assertion in a particular form.
- **Axioms:** assertions (including rules) in a logical form that together comprise the overall theory that the ontology describes in its domain of application. This definition differs from that of "axioms" in generative grammar and formal logic. In those disciplines, axioms include only statements asserted as a priori knowledge. As used here, "axioms" also include the theory derived from axiomatic statements.
- **Events:** the changing of attributes or relations [5].

6. Ontology Engineering

Ontology engineering (or ontology building) is a subfield of knowledge engineering that studies the methods and methodologies for building ontologies. It studies the ontology development process, the ontology life cycle, the methods and methodologies for building ontologies, and the tool suites and languages that support them.

There is no one correct methodology for developing ontologies.It is usually an iterative process. We can start with a rough first pass at the ontology and then revise and refine the evolving ontology. Ontology is a model of a real domain in the world and the concepts in the ontology must reflect this reality. After defining an initial version of the ontology, we can evaluate and debug it by using it in applications or problem-solving methods or by discussing it with experts in the field. As a result, we will almost certainly need to revise the initial ontology. This process of iterative design will likely continue through the entire lifecycle of the ontology.

Developing Ontology may include:
- Selection of Domain and Scope.
- Consider Reuse.
- Find out Important Terms.
- Defining Classes and Class Hierarchy.
- Defining Properties of Classes and Constraints.
- Create Instances of classes [6].

7. Ontology Building Tools

7.1 Protégé:is an ontology and knowledge base editor produced by Stanford University. Protégé is a tool that enables the construction of domain ontologies, customized data entry forms to enter data. Protégé allows the definition of classes, class hierarchies, variables, variable-value restrictions, and the relationships between classes and theproperties of these relationships. Protégé is free and can be downloaded from http://protégé.stanford.edu [7].

Protégé comes with visualization packages such as OntoViz, EZPal, etc.; all of these help the user visualize ontologies with the help of diagrams. Stanford University is doing a magnificent job of continually improving Protégé. As part of its last update, Protégé now includes an interface for SWRL (Semantic Web Rule Language), which sits on top of OWL to do math, temporal reasoning, and adds Prolog-type reasoning rules. Stanford has a tutorial that covers the basics of using Protégé with the OWL plug-in.

7.2 ISAVIZ: IsaViz is a visual environment for browsing and authoring RDF models as graphs. This tool is offered by W3C Consortium. IsaViz [7] was developed by Emmanuel Pietriga.The first version was developed in collaboration with Xerox Research Centre Europe which also contributed with XVTM, the ancestor of ZVTM (Zoomable Visual Transformation Machine) upon which IsaViz is built. As of October 2004, further developments are handled by INRIA Futurs project In Situ. IsaViz also includes software developed by HP Labs (Jena 2 Semantic Web Toolkit), the Apache Software Foundation (Xerces Java 2), and makes use of the GraphViz library developed by AT&T Research. IsaViz does not follow or include any methodology for building ontology. IsaViz imports RDF/XML and N-Triples, and exports RDF/XML [13], N-Triples, Portable Network Graphics (PNG) and Scalable Vector Graphics (SVG). Therefore, it is possible to import ontologies to other editors, for instance, Protégé or OilEd. The IsaViz environment is composed of four main windows: the IsaViz RDF Editor window, the Graph window, the Definition window and the Attribute window [8].

7.3 APOLLO: is a user-friendly knowledge modeling application. The modeling is based around the basic primitives, such as classes, instances, functions, relations etc. Internal model is built as a frame system according to the internal model of the OKBC protocol.

Apollo's class system is modeled according to the OKBC. The knowledge base consists of ontology's that are hierarchically organized. Ontology can inherit other ontology's and then use classes of inherited ontology's as its own. Every ontology inherits at least one ontology − a default ontology, which contains all primitive classes: Boolean, integer, float, string, list etc. Class contains slots of two types: non template and template slots.

Apollo currently does not support non template class slots. For each class is possible to create a number of instances. An instance inherits all slots of the class. Each slot has a set of facets [9].

7.4 SWOOP: is a Web-based OWL ontology editor and browser. SWOOP contains OWL validation and offers various OWL presentation syntax views. It has reasoning support and provides a multiple ontology environment. Ontologies can be compared, edited and merged. Different ontologies can be compared against their Description Logic-based definitions, associated properties and instances.

SWOOP's interface has hyperlinked capabilities so that navigation can be simple and easy. SWOOP does not follow a methodology for ontology construction. Users can reuse external ontological data. This is possible either by purely linking to the external entity, or importing the entire external ontology. It is not possible to do partial imports of OWL. There are several ways to achieve this, such as a brute-force syntactic scheme to copy/paste relevant parts (axioms) of the external ontology, or a more elegant solution that involves partitioning the external ontology while preserving its semantics and then reusing (importing) only the specific partition as desired.

It is possible to search concepts across multiple ontologies. SWOOP makes use of an ontology search algorithm, that combines keywords with DL-based in order to find related concepts. This search is madealong all the ontologies stored in the SWOOP knowledge base [10].

8. Ontology Projects

In the next part, we will highlight some of the Ontology projects in the field of Holy Quran Knowledge, which is one of active areas in the ontology Development, besides other domains mentioned in the abstract of the paper.

The three common ontologies covering Al-Quran are: **Quranic topics**, **Arabic Quran Corpus**, and **QuranA**[11].

The Quranic-topics ontology contains nearly 1,100 Quranic concrete and abstract concepts linked to all verses of Al-Quran. Use existing Quranic concepts from the Islamic scholarly book Mushaf Al Tajweed. These concepts in the index have an aggregation relationship; the hierarchy of concepts is non-reflexive, nonsymmetric, and transitive [12].

Arabic Quran Corpusextracts 300 concepts and 350 relations from Al-Quran. The relationship types connecting concepts using predicate logic are Part-of and IS-A. The ontology is based on a famous book describing Al-Quran called 'TafsirIbnKathir' [13].

QuranA, ontology for Al Quran in the scope of pronoun antecedents. This ontology consists of 1,050 concepts and more than 2,700 relations. Additionally, the relationship types connecting concepts are has-antecedent, has-concept and has–a-segment [14].

9. Conclusion

Ontology defines a common vocabulary for researchers who need to share common understanding of the structure of information among people or software agents and enable reuse of domain knowledge. Furthermore, it includes machine-interpretable definitions of basic concepts in the domain and relations among them.Thus, it makes domain assumptions explicit while separate domain knowledge from the operational knowledge.

References

[1] Tom Gruber, "Ontology", Stanford University, 2008.
[2] FlorianoZini , Leon Sterling "Designing Ontologies for Agents", Department of Computer Science Software Engineering,University of Melbourne, Parkville 30502,Australia,1999.
[3] Michael Wooldridge," An Introduction to Multiagent Systems", University of Liverpool, 2008.
[4] Catherine Roussey, Francois Pinet, Myoung Ah Kang, and Oscar Corcho, "An Introduction to Ontologies and Ontology Engineering".
[5] Matthew Horridge " A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools" , Edition 1.3.
[6] BhaskarKapoor, Savita Sharma, "A Comparative Study Ontology Building Tools for Semantic Web Applications", International journal of Web & Semantic Technology (IJWesT), Vol.1, Num.3, July 2010
[7] http://protege.stanford.edu
[8] http://www.w3.org/2001/11/IsaViz/Overview.html
[9] http://apollo.open.ac.uk/index.html.
[10] AdityaKalyanpur, BijanParsia, EvrenSirin, Bernardo Cuenca Grau, James A. Hendler: "Swoop, "A Web Ontology Editing Browser", J. Web Sem. 4(2): 144-153 (2006).
[11] Sameer M. Alrehaili, Eric Atwell," Computational ontologies for semantic tagging of the Quran: A survey of past approaches", School of Computing, University of Leeds, Leeds LS2 9JT, UK.

[12] N. Abbas," Quran search for a Concept Tool and Website", Unpublished Dissertation, university of Leeds, 2009.

[13] K. Dukes, "Statistical Parsing by Machine Learning from a Classical ArabicTreebank", School of Computing, University of Leeds, 2013, Available at: http://corpus.quran.com/.

[14] A. B. Muhammad," Annotation of Conceptual Co-reference and Text Mining the Qur'an", 2012.

Author Profile

EimanAlsiddigAltayeb Ibrahimreceived the B.S. degree in Computer science: Omdurman Islamic University in 2007, M.S.C degree in Computer Science: Sudan University of Science and Technology in 2014, PhD researcher in computer science at Sudan University of Science and Technology. Now works for Omdurman Islamic University.

Mohammed Awad Mohammed Ata Elfadielreceived the B.S. degree in Computer science: Omdurman Islamic University in 2007, M.S.C degree in Computer Programming: Sudan Academy of Science in 2011, PhD in computer science: Alribat National University in 2015. Now works for Open University of Sudan.