

Exploring Hybrid Methodologies for Ontology Development: An Extension of Software Engineering Process Models

Dr. Leela Vyas, Dr. Rohan Kaur, and Dr. Aisha Al-Mannai

Dr. Leela Vyas and Dr. Rohan Kaur, Department of Information Systems, University of Melbourne, Australia; Dr. Aisha Al-Mannai, College of Computer Science and Engineering, Qatar University, Qatar

Abstract

Indexed keywords: Computer Engineering, Advanced Computing, Technology, Open Access

Article History: Received: 14 August 2022 | Accepted: 03 November 2022 | Published: 19 November 2022

W3C's Semantic Web intends a common framework that allows data to be shared and reused across application and enterprise. The semantic web and its related technologies are the main directions of future web development where machine-processable information which supports user tasks. Ontologies are playing the vital role in Semantic Web. Researches on Ontology engineering had pointed out that an effective ontology application development methodology with integrated tool support is mandatory for its success. . Potential benefits are there to ontology engineering in making the toolset of Model Driven Architecture applicable to ontology modeling. Since Software and Ontology engineering are two complimentary branches, the scope of extension of the well proven methodologies and UML based modeling approaches used in software engineering to ontology engineering can bridge the gap between the engineering branches. This research paper is an attempt to suggest an exclusive hybrid methodology for ontology development from existing matured software engineering. Philosophical and engineering aspects of the newly derived methodology have been described clearly An attempt has been made for the application of proposed methodology with protégé editor. The full-fledged implementation of an domain ontology and its validation is the future research direction.

Network Protocols, Wireless Network, Mobile Network, Virus, Worms & Trojan

1. INTRODUCTION

The semantic web vision aims to integrate and convert the vast amount of information available on the internet into a machine-understandable network. This enables the knowledge sharing and its reusability across domains. It is a mesh of information which can be linked in a way that can easily be processed by machines and which can produce resources processed by machines [1]. Ontologies are playing the vital role behind the success of emerging semantic web. According to Gruber's definition, ontology [2] is the "formal specification of a conceptualization". Ontologies form a global



pool of reusable, shared knowledge resources. It constitutes a special kind of software artefact conveying a certain view of the world (conceptualization), specifically designed with the purpose of explicitly expressing the intended meaning of a set of agreed existing objects [3]. It aims to produce technologies capable of performing reasoning on semi-structured information [4]. Mizoguchi

DOI:10.5121/ijitcs.2016.6104

37

summarized the merits of ontology as follows: “Ontology provides a common vocabulary, and an explication of what has been often left implicit”. According to Mizoguchi, the linked data, systematization of knowledge and standardization constitute the backbone of knowledge within a knowledge base system. Ontology engineering is a challenging and dynamic research field, which deals with the methods, methodologies and tools for building and managing ontology. This branch of engineering aims at making explicit knowledge contained within software applications, enterprises and business procedures for a particular domain. Ontology engineering offers a direction towards solving the inter-operability problems brought about by semantic obstacles, i.e. the obstacles related to the definitions of business terms and software classes [5]. Matured software engineering has been established on modeling language and well proven process models which are lacking in a considerable fashion in ontology engineering which is causing large scale enterprise application development involving the semantic web technologies. The existing ontology development methodologies provide only an ad hoc approach to bridge the above mentioned gap with limited functionality and performance. Researchers of ontology engineering domain came up with ideas and proposals through research publications and workshops in international conferences on ontology engineering to address this issue. IBM Research group came up with an approach-Eclipse Modeling Framework-Based Ontology Engineering System (EODM) by leveraging Model Driven Architecture (MDA) and Ontology Definition Model (ODM), which enable model transformation [6]. The main goal of this research paper is to propose an exclusive methodology for ontology development compactable to UML models from the well proven methodologies of software engineering to bridge the gap between two complementing engineering branches as well as to leverage the proven tool/set of software engineering to ontology Engineering.

2. RELATED WORK AND BACKGROUND OF RESEARCH

2.1. Ontology and Model Driven Architecture

Object Management Group (OMG)’s Model Driven Architecture (MDA)- an industry body devoted to the development and promotion of standards for enterprise computing which is based on Unified Modeling Language (UML) and related standards. MDA defines an architecture that provides a set of guidelines for structuring specifications expressed as models. UML is one of the well proven modeling languages and is supported by widely-adopted CASE tools and its UML things can be extended with new concepts and adapted to specific modeling needs. “Ontology is a specialization of a conceptualization”. This author agrees with the idea presented in [5], that a conceptual modeling language would be able to express semantic relations and appropriate integrity

constraints which help to ensure that instances of objects and property values are semantically "valid". From a conceptual model representation one may generate graphical models for visualization. This approach leverages the specification of Ontology Definition Meta model (ODM). Designing ontology is a non trivial task that requires sophisticated tool support. Ontology languages are based on (description) logics, and thus tool support can and must take into account the underlying semantics of ontology. The traditional ontology management systems provide only ad hoc and incomplete approach towards model interoperability. MDAbased tool to convert ontologies expressed in any language having a metamodel defined used the OMG's Meta Object Facility (MOF) model to an equivalent representation in Resource Description Framework (RDF) but with the same metamodel. MOF tools use metamodels to generate code for managing models and metadata. Therefore in March 2003, the OMG issued an RFP for an Ontology Definition Metamodel (ODM) [6], which seeks a specification of a MOF 2.0 (MOF2) compliant metamodel, a UML 2.0 profile and any additional information needed to support the development of ontology using UML modeling tools. UML modeling will enable its compatibility on implementation of ontology in the W3C Web Ontology Language (OWL) as well as the forward and reverse engineering of ontology.

2.2. Need of a Methodology

The development of large-scale enterprise applications has become increasingly complex due to the massive growth of enterprise data and the constant changing of requirements. Semantic technology has been seen as a crucial alternative for managing this complexity by providing a solid and flexible infrastructure for information exchange, retrieval, sharing, and discovery. Since ontology is playing the crucial role in providing semantic technology solutions, it is mandatory to standardize the ways ontology is developed. Development of ontology is more complex compared to any other kind of typical software project due to various factors such as the necessity of sophisticated tool support, heterogeneous platforms, dynamic changes on business needs, lack of performance engineering, low failure tolerance etc. An effective formal development methodology only can guarantee a solution to the problems pertaining to software development. One of the reasons for the success as well as the maturity of software engineering itself is the availability of appropriate off-the-shelf and on-the-shelf methodologies. These include classical sequential methodologies, RUP suits for iterative development and agile methodologies supports development with stake holders. There is no one correct methodology for developing ontology, since there is no one correct way to model a domain [7]. There are few methodologies by Uschold and King, Gruninger and Fox were introduced but failed to name a standardized one. Ontology itself is a data model based on formal logic and greatly overlaps with a UML object model, as both share many basic concepts. Model Driven Ontology [8], a new methodology was developed later using UML object model as the common means for expressing ontology models. The rigorous development process for ontology building requires the use of methodologies and platforms more or less equivalent to software development. A methodology with fewer curves for software engineers can definitely make ontology development appropriate for business users. A software engineering approach to Ontology building (UPON) [9] has been proposed based on a rich set of resemblances between software engineering and ontology engineering in terms of stages and phases. UPON methodology focused on

exploiting the possibilities of Unified Process (UP) and Unified Modeling Language (UML). UPON is a novel approach for large scale ontology development that recommends an iterative life for ontology development by leveraging the features of UP and UML. However, the lack of an agile methodology feature in UPON and the resulting complexity make it unsuitable for effective ontology development.

3. PROPOSED METHODOLOGY

3.1. Background of Proposed Methodology

In this section, I will explain the core concept behind the proposed ontology development methodology. Ontology development methodologies mainly prescribe guidelines for the specification, conceptualization, formalization and implementation of ontology [10]. The proposed ontology development methodology was derived as a hybrid of two well proven software process models, traditional linear waterfall and incremental, iterative Rational Unified Process (RUP). The proposed ontology development methodology's stages are originated from the lifecycle proposed by the METHONTOLOGY, [11] a methodology proposed for ontology construction by the Foundation of Intelligent Physical Agents (FIPA), which promotes interoperability across agent-based applications. The engineering behind the METHONTOLOGY is the intermediate representation in terms of different models such as specification model-semiformal specification using a set of intermediate representations ,conceptual model and a formalized model (e.g., Description Logic Ontology UML Profile) which will be implemented in an ontology implementation language (e.g., Web Ontology Language (OWL) or Resource Description Framework (RDF).

The proposed approach deals with the various stages of ontology development tuned to linear approach and later the overall development methodology fitted under Rational Unified Process (RUP). RUP has been chosen as the root methodology to derive the proposed one due to its ability to support iterative and incremental approach. The proposed methodology classifies the phases into three such as pre- development, development and post development. Pre development stage is concerned with feasibility study of ontology domain that includes assessing the scope of domain with clear definition of boundaries. Development stage is with the main goal of producing essential model. The last stage is post development where implementation model definition is constructed. Every phase delivers a specific output with the common goal of creating functional component based on ontology that can be used in several systems and scenarios. In pre-development stage the output is feasibility report that is used in subsequent phase development to construct essential model definition. The latter artefact represents the schema of problem domain that has to be coupled with instances from the real world. This is conducted in the last stage post- development which produces functional component for usage in various systems. Core stages of ontology development phases are briefed below.

3.2. Linear Model of Development Stages

Feasibility Analysis. This stage includes assessing the scope of the project with a clear definition of boundaries. Domain experts will participate in this phase as agile methodologies ensure their involvement across the planning, design and development phases.

Domain Vocabulary Acquisition. The acquisition of domain vocabulary is the key step for further definition of ontology. Concepts/Classes are the focus of most ontology. These classes are derived as a result of domain vocabulary acquisition. Automated tools/mind mapping tools can be used at this stage for large scale ontology development.

Enumeration of Concepts and Properties. This stage leads to the derivation of both Data properties and Object properties. Concepts makes a semantic translation from one source into another possible. Properties and attribute values will help to achieve the needed shared vocabulary. Properties and their values are playing vital role in the Individual instances of concepts

Taxonomy Identification. This stage defines the concept hierarchies. A top-down development process starts with the definition of the most general concepts in the domain and subsequent specialization of the concepts. A bottom-up development process starts with the definition of the most specific classes, the leaves of the hierarchy, with the subsequent grouping of these classes into more general concepts. A combination development process is a combination of the topdown and bottom-up approaches:

Adhoc Binary Relationships. This stage establishes the appropriate semantic/structured relationship among the identified class hierarchies. Organizing the classes into a hierarchical taxonomy based on the generalization principle that an instance of a subclass, will necessarily be an instance of the superclass. If a class A is a superclass of class B, then every instance of B is also an instance of A.

Describe Concepts attributes and Relationships. This stage describes the internal structure of the concept. Most of the remaining terms excluded from the class list after the domain vocabulary acquisition stage, are likely to be properties of the classes. These properties become slots attached to classes. This stage also includes the relationships between individual members of the class and other items

Add Complex Restrictions and Rules. This stage describes the value type, its allowed values, the number of the values (cardinality), and other features of the values the slot can take. For example, the value of a name slot is one string; that is, name is a slot with value type string. A slot can have multiple values and the values are instances of the class.

Vocabulary Linking with Data. The last step is creating individual instances of classes in the hierarchy. Defining an individual instance of a class requires choosing a class, creating an individual instance of that class, and filling in the slot values. At this stage the vocabulary is absolutely linked with the real data. In the proposed agile methodology, the ontology development stages fit into the traditional linear Waterfall process model as shown in figure 1.

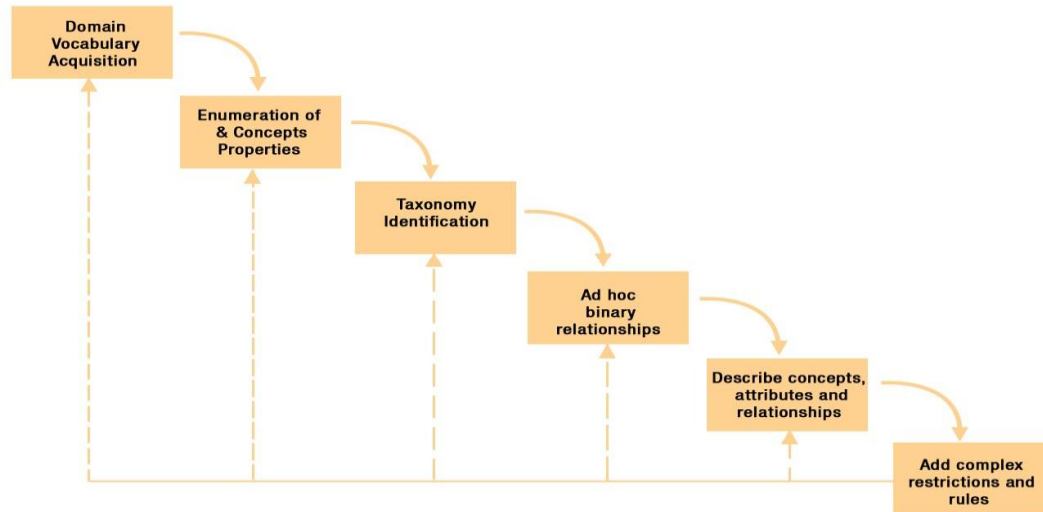


Figure 1. Development stages into linear model

3.2. Framework of Proposed Ontology

The final framework of the proposed methodology fits the various stages of ontology development into the phases of incremental and iterative development methodology, RUP. This provides a disciplined approach to assigning tasks and responsibilities within a development team. The Rational Unified Process captures many of the best practices in modern software development in a form that is suitable for ontology development too. The phases of proposed ontology development methodology along with their stages are fitted into RUP phases, Inception, Elaboration, Construction and Transition. The overriding goal of the inception phase is to achieve concurrence among all stake holders on life-cycle objectives for the project and can map to feasibility study. The purpose of the elaboration phase is to analyze the problem domain, establish a sound architectural foundation, develop the project plan, and eliminate the project's high risk elements. During the construction phase, all components and application features are developed and integrated to the product, and all features are thoroughly tested. These objectives are well mapped to the ontology definition phase. Finally the ontology implementation is mapped towards transition phase. Figure 2 illustrates the final framework of the proposed ontology development methodology.

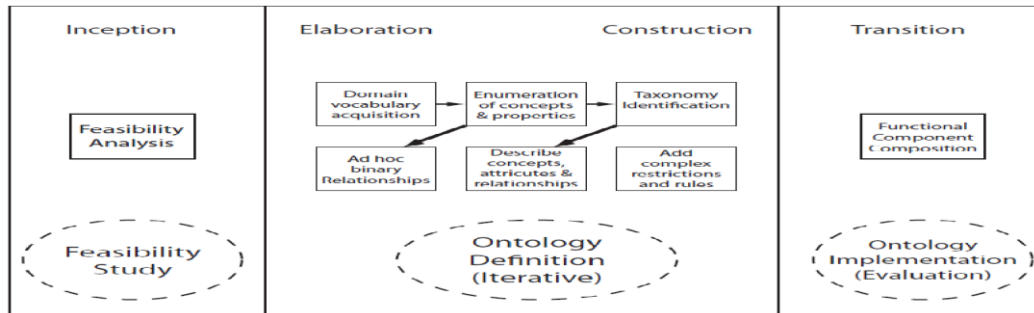


Figure 2. Final Framework of Proposed methodology

4. APPLICATION SCOPE OF THE METHODOLOGY

The proposed methodology can be applied for the development of domain Ontologies. With Protégé ontology editor, a partial attempt has been made for a prototype development on Pizza domain. For the prototype development, popular pizza ontology been considered as the base model. The requirement was to develop an ontology that contains basic information about Pizza. Resource Description Framework (RDF) was chosen as the implementation language.

RDF codes were generated with Protégé 4. A complete feasibility analysis was conducted on

Pizza domain which helps to identify the concepts- generic classes Pizza, PizzaBase and PizzaToppings. These identified classes directed to the Ontology definition stage in which hierarchy of classes was created. More subclasses were identified and taxonomy relationships were introduced. As an iterative manner, concepts and attributes were introduced to the classes. Iterative approach taken over from RUP methodology helps here to add complex restriction and rules along with object properties on every iteration. An incremental approach has been followed within the ontology definition phase. Proposed ontology development methodology provides the opportunity to follow the incremental and iterative approach. Ontology implementation is yet to complete but will follow functional component competition. Figure 3 is a screen shot taken during the prototype development.

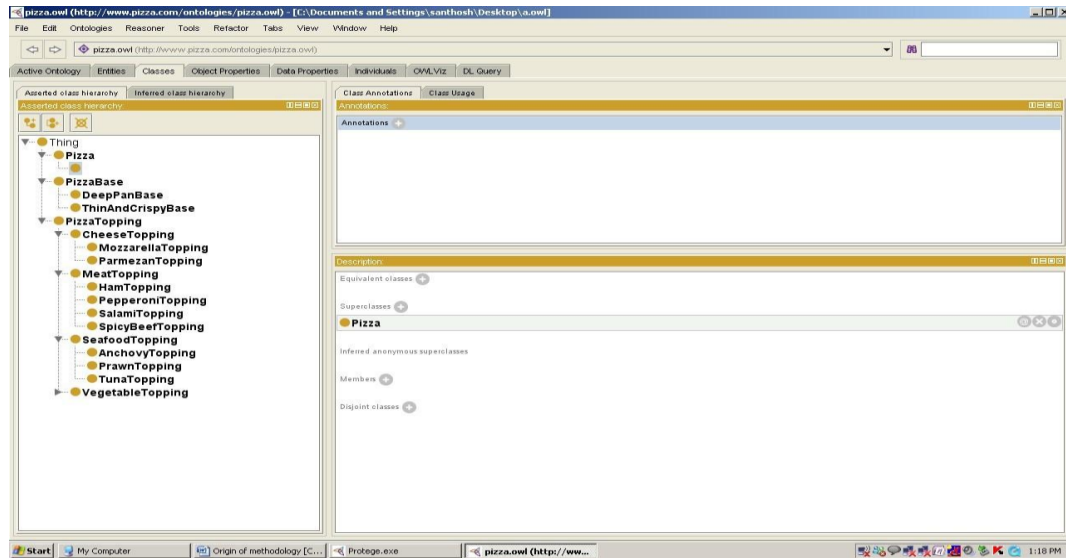


Figure 3. Screen shot of prototype development

5. CONCLUSION

This paper has investigated the scope of ontology within the context of the upcoming semantic web technologies. The research on this domain confirmed that the key player of semantic web technology is ontology. Even though ontology playing a crucial role on the mentioned technology, there was no specific methodologies for ontology development with tool support compactable with well proven software methodologies. This paper examined the scope extending UML modeling to ontology development within the context of Model Driven Architecture and Web Ontology Language (OWL). The core of this research paper is the derivation of an exclusive methodology for ontology development. The proposed methodology was derived as a hybrid of linear model of waterfall and iterative Rational Unified Process. Various stages of ontology development were well mapped to RUP phases. My future work is to develop full-fledged domain ontology on the basis of the proposed methodology to evaluate its efficiency as well to identify the bottlenecks of proposed methodology.

REFERENCES

- [1] Bittencourt, E. de Barros Costa, M. Silva, and E. Soares, "A computational model for developing semantic web-based educational systems," *Knowledge-Based Systems*, vol. 22, no. 4, pp. 302 – 315, 2009.
- [2] T.R. Gruber. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In Roberto Poli Nicola Guarino, editor, *International Workshop on Formal Ontology*, Padova, Italy, 1993. Technical report KSL-93-04 Knowledge Systems Laboratory, Stanford University.
- [3] O. Mendes. A. Aban "Software Engineering ontology: A development methodology" <http://www.gelog.etsmtl.ca/publications/pdf/839.pdf>, accessed on july 2014
- [4] Ahmed Z, "Web to Semantic Web and Role of Ontology in its Development" Ahmed, Zeeshan. (August 1992). *Web to Semantic Web and Role of Ontology in its Development*. Semantic webjournal [Online]. Available: <http://www.semantic-web-journal.net/sites/default/files/swj99.pdf>

- [5] Line Pouchard, Nenad Ivezic and Craig Schlenoff (2000) "Ontology Engineering for Distributed Collaboration in Manufacturing" to appear in the Proceedings of the AIS2000 conference, March 2000.
- [5] Ramberg, Jan. ICC Guide to Incoterms 2000. Understanding and Practical Use; International Chamber of Commerce 2000
- [6] "An MDA-Based Software Development Environment for Ontology Engineering" (November 2005)
[http://domino.research.ibm.com/library/cyberdig.nsf/papers/OD884EF43842DFD8852570BD0060E974/\\$File/rc23795.pdf](http://domino.research.ibm.com/library/cyberdig.nsf/papers/OD884EF43842DFD8852570BD0060E974/$File/rc23795.pdf)
- [7] Dragon Gasevic, Dragan Djuric, and Vladan Devedzic, "Model Driven Architecture and Ontology Development", Springer, 1st edition, July 2006.
- [8] <http://ceur-ws.org/Vol-440/paper14.pdf>, Model Driven Ontology, A new methodology for Ontology Development.
- [9] A.D. Nichola, M.Missikoff, R.Navigli "A software engineering approach to ontology building," Science Direct Trans. Information Systems, vol. 34, pp. 258-275, Jan. 2009.
- [10] Mario C.C., Ontologies for Software Engineering and Software Technology, Springer, 2004.
- [11] A. Gómez-Pérez, M. Fernández-López, and O. Corcho, ,Ontological Engineering. Berlin: Springer,