Management Research and Practice
Volume 13, Issue 4 / December 2021

A STUDY ON PRACTICAL APPROACHES IN BUILDING DOMAIN ONTOLOGIES

Liliana DOBRICA
University POLITEHNICA of Bucharest, Romania
liliana.dobrica@aii.pub.ro

Abstract
This article presents a study of several methodologies for building ontologies. It focuses on approaches where Protégé tool is used to develop an ontology appropiate for a defined purpose. The most relevant aspects are identified with the objective of managing the good practices when solving particular domain challenges. The study discusses various characteristics of ontology development methodologies based on Protégé and analyses their specific constraints. An important outcome of the study is that the evolution of software engineering processes influences the evolution of ontology development approaches. The global and agile aspects of the development activities are among the common properties of both. The main contributions described in this article are related to finding out a variability in the development activities and their flow of control, practical illustration of processes of manual or automated classification in an ontology, how ontologies are utilized in applications. The study aims to act as an introductory guide for ontology developers in their beginning stage.
Keywords: ontology, ontology engineering, knowledge engineering, knowledge representation, protégé

1. INTRODUCTION

The globalization and the Internet are part of the of the new business ecosystems reality with a growing need of better communication infrastructure, interaction and interoperability. Efficient solutions are necessary to support better knowledge integration, sharing and reuse. A key trend in the recent years is standardization to support the uniformity of resources and systems and the possibility of their reuse (Dobrica, 2018). Business organizations, interacting in an ecosystem as actors, need to be competitive to improve their hardware, software and information technologies. The use of ontologies for the development of the university educational ecosystem as an element of the information technology infrastructure is a recent proposed solution of improvement in the research community (Pustovalova et al., 2021). Blended or distance learning formats have been considered and analyzed in the past (Colesca et al. 2009) under perspective of students, that are consumer of the university educational services. Mostly in the face of the 2020 pandemic, educational organizations should accelerate introducing and improving digital tools and information systems for organizing remote interaction between all actors in the educational ecosystem. Several groups of actors including educational organizations, consumers of the educational services, business communities, integrators of education, science and business, regulatory and administrative authorities are interacting in an educational ecosystem and their needs and requirements should be identified and included in the context of the development process.

A study of practical approaches in building domain ontologies focuses on the exploration of the most recent methodologies where Protégé tool is used to develop an ontology appropiate for a defined purpose. The goal is to identify the most relevant aspects with the objective of understanding and applying good practices when solving particular domain challenges (Dobrica and Niemela, 2008) (Dobrica and Pietraru, 2017). The study compares Protégé-based ontology development methodologies and analyses their specific constraints of the context, case studies requirements, or types of application projects.

The article is structured in several sections. It begins with a background section, where discusses about ontology various definitions and their interpretations. Also in this section are introduced key elements of methodologies in ontology engineering. The next section presents relevant approaches focussing on a general perspective having the central element Protégé IDE. Finally, a last section is a practical description of tools of Protégé IDE in ontologies development activities.

ISSN 2067- 2462

Management Research and Practice
Volume 13, Issue 4 / December 2021

2. BACKGROUND

2.1. Ontology Definition

Several definitions have been proposed during the years. The term ontology has been introduced in (Schreiber et al, 1994) to denote „a specification of a conceptualization" in the context of the domain knowledge modeling with the CommonKADS methodology. A domain knowledge model distinguishes two types of ontologies, the model ontology and the domain ontology. Many other scientists discussed about a better ontology definition. In (Gruber, 1995) the notion of ontology is refined in a "shared and formal specification of a conceptualization".

An interesting historical perspective behind the notion is presented in (Gutierrez and Sequeda, 2021), where essential elements involved in the notion of knowledge graphs roots are traced to history periods in a data and knowledge presentation framework. In a similay way (Hitzler, 2021) contributes with a review of the semantic web field where ontology is a key concept in the beginning of this field of research during the first decade of 2000s. According to a many cited source from (Gruber,1993) „an ontology is a formal, explicit specification of a shared conceptualization". Research community argues that this definition still needs interpretation and is rather generic. Recently it has been agreed on the following *"An ontology is an explicit specification of a shared conceptualisation that holds in a particular context"*(Hoekstra, 2017). Explicit means a structured language, explicit definitions, not necesarily formal. Shared means a consensus on the definitions, shares openly (on the web/on the global enterprise). Conceptualization is associated to use of concepts, facts and relations. And, finally, the context is the one that connects the building of the ontolgy to a particular purpose, domain or task.

In a Semantic Web context, ontologies are a main vehicle for data integration, sharing, and discovery, and a driving idea is that ontologies themselves should be reusable by others. They represent an essential component to developing the web of data and semantic web applications. The perspective of knowledge sharing and reuse facilitates communication between people or interoperability at machine level, reuse of domain knowledge or make domain knowledge explicit and analyse domain knowledge.

The scientific community developped several types of ontologies such as top ontologies, generic ontologies, core ontologies or domain ontologies. It isn't a goal in itself building an ontology considering that an ontology should require the minimal ontological commitment sufficient to support the intended knowledge sharing activities. This article focuses on the research approaches for building domain ontologies. In this way an ontology is a conceptual model of an observed reality and, in essence, is a repository of interlinked concepts pertaining to a given application domain In this more precise sense an ontology is a knowledge base of concepts, properties and their relationships, specified in a knowledge representation language based on a formal logic.

2.2. Methodologies in ontology engineering

A methodology consists of a number of elements that can be depicted graphically in the form of a pyramid (Scriber et al., 2000). The methodological pyramid has five layers, where each consecutive layer is built on top of the previous one (Figure 1). The lowest layer is the "worldview" of the methodology. These are the advertising slogans of an approach. These slogans need to be grounded in theory, methods, tools and practical case studies which constitute the other four layers. The slogans can be formulated as a number of principles that are based on the lessons learned in the past.

A high-quality ontology requires a rigorous, systematic engineering approach. Since the 1990s, many researchers have begun to propose various approaches to build ontologies based on a development methodology. In the beginning the ontology development has been considered a creative activity that could be seen more an art than a systematic repetitive engineering process. The ontology engineering is a concept emerged from knowledge engineering (Dobrica, 2021). Methods in software engineering have been adopted by many creators of knowledge engineering (Scriber, 2000) and ontology engineering methodologies. For instance, CommonKADS, a mature methodology in the knowledge engineering, has been inspired by software engineering (Studer,1998) (Schreiber, 2000) activities. Development, support and management are three

Management Research and Practice
Volume 13, Issue 4 / December 2021

types of activities included in the engineering process of ontologies. Among the development activities it could be mention procedures to specify, conceptualize, formalize, and implement an ontology.

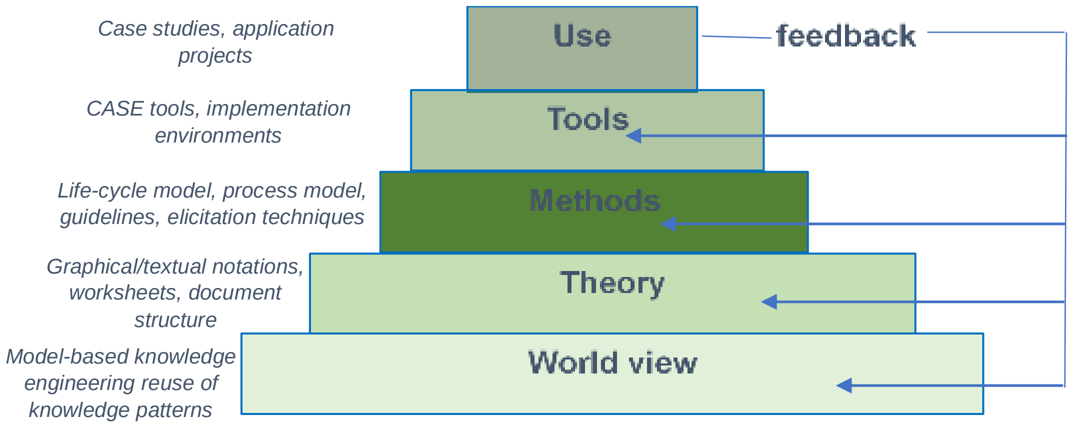
Case studies, application projects
CASE tools, implementation environments
Life-cycle model, process model, guidelines, elicitation techniques
Graphical/textual notations, worksheets, document structure
Model-based knowledge engineering reuse of knowledge patterns
Use
feedback
Tools
Methods
Theory
World view

FIGURE 1 - THE METHODOLOGICAL PYRAMID (ADAPTED FROM (SCHREIBER ET AL.,2000)).

Since the 1990s, researchers have begun to build ontology engineering methodologies. Among the initial ontology development methods are IDEF5 (Menzel and Mayer, 1992) or METHONTOLOGY (Fernández-López et.al., 1997). Some of the 2000 first decade methodologies include On-To-Knowledge (Sure et al., 2004), the Ontology Development 101 (Noy and McGuinness, 2001) and UPON (De Nicola et al., 2009). The second decade of 2000 has methodologies such as UPON Lite (De Nicola and Missikoff, 2016) or YAMO (Dutta, et al., 2015).

Ontology engineering helps domain knowledge experts better understand their own business reality by systematically exploring and describing it.

3. RELEVANT APPROACHES

This section discusses about relevant approaches building ontologies with Protégé. Figure 2 presents a general view where knowledge domain experts and knowledge analysts collaborate in the knowledge capture process from various case and research studies.

The context of using knowledge in the organization business process and its management is presented in Figure 2. The Protégé tool is used build and implement an ontology (Ontology – Tbox) and, then to insert and populate it (Ontology – ABox) with individuals (concepts instances). The outcome of this process is the knowledge base. Thus, real situations are used to build the ontology Tbox, and, furthermore, structured knowledge is added and populate the ontology ABox through knowledge experts in the application domain.

Several approaches are presented bellow. They have been selected from scientific databases looking for the year of publication after 2015 and variability in purposes, case studies and the sequence of steps.

(Dhingra and Bhatia, 2015) presents the development of ontology in laptop domain for knowledge representation. The basic steps of the development of this ontology are: determine the domain knowledge, determine the key concepts, build the taxonomy of classes, identify the relationships between classes, check the consistency and implementation of the ontology in the application. Three ontologies are developed in this domain, namely, for specification, selling and review.

(Dutta et al., 2015) introduces YAMO for large scale faceted ontology construction. It includes ten steps and is applicable to build any domain ontology to be shared in a wide community. The steps are: domain identification, domain footprint, knowledge acquisition, knowledge formulation, knowledge production, term standardization, knowledge ordering, knowledge modeling, knowledge formalization, and evaluation. It refers to a set of guiding principles to ensure the quality of the ontology. Ontologies become knowledge graphs and findable, accessible, interoperable, reusable (FAIR) principles (Wilkinson et al., 2016) have been recently considered.

Management Research and Practice
Volume 13, Issue 4 / December 2021

(de Nicola and Missikoff, 2016) introduces a methodology for rapid ontology engineering UPON Lite. It is a simple, agile ontology engineering method intended to place end users at the center of the process. This method is derived from the UPON methodology. It is organized as a sequence of steps, where the outcome of each one is enriched and refined in the succeeding step; the steps produce the following outcomes: lexicon, glossary, taxonomy, predication, parthood and ontology. Some of these steps can be performed in parallel.

(Schachinger, et. al, 2016) presents an ontology abstraction layer for smart grid interaction in building energy management systems. OD 101 methodology is chosen due to its simplicity and clear structure. It defines the scope of the ontology, the reuse, the most important terms. It builds the most the class hierarchy and it defines in parallel the properties of classes, it specifies data properties and domains and ranges for object properties, and in the end instances.

(Ungkawa et al, 2017) presents a methodology for constructing form ontology. It is considered a simple methodology with the following characteristics: incremental/iterative, lightweight/agile, documentation, evaluation (application based), maintenance, using ontology language OWL2, based on a complete tool: Protégé: builder, reasoner, and visualizer.

(Shrivastava et al., 2018) discusses about an ontology development for university. This approach follows the steps: scope and purpose of ontology; intended users for the ontology; developing competency questions; identification of groups; generating frequencies; implementation of ontologies; evaluations. A set of sub-steps are included in the implementation after terms are identified and their frequencies are generated and the main concepts are identified on the basis of the highest frequency on a particular sub area. The implementation include conceptualization, identification of relations, creation of instances and visualization.

(Ciancarini et al., 2019) presents an ontology of software relational factors from financial services. The design considers reusing ontology design patterns (ODPs) according to an extension of the eXtreme Design methodology (XD). The reuse of ODPs speeds up the ontology design process, eases design choices, produces more effective results in terms of ontology quality, and boosts interoperability. The ontology guarantees interoperability by keeping the appropriate alignments with the external ODPs and provides extensions that satisfies more specific requirements. This approach is iterative and incremental involving a design team, a testing team and a customer team representing domain experts. Requirements recorded as stories as in agile development, are transformed into competency questions (CQs). CQs are converted into SPARQL queries that are executed for the ontology validation. The design is modular on CQ basis and reuses ODPs. Integration testing of the ontology validates its logical consistency and the ability to detect errors by deliberately introducing contradictory facts. The XD methodology description is modeled with an UML activity diagram. The approach is successful in linked open data projects.

(Maduray et al., 2020) presents an ontology for the supply and generation of electricity. OD101 methodology which proposes an iterative process for building ontology is applied. It includes data collection from various sources of relevant terminologies and it creates a list of axioms, which are sentences that explain the domain and provide the key concepts and relationships between them in a natural language. Next is the formalization of ontology using two logical languages Description Logic (DL) and Web Ontology Language (OWL). Modeling of ontology creates the lists of concepts, properties and instances from the axioms. Logical modelling of the axioms of the domain uses DL representation. It acknowledges the reuse of existing ontologies.

(Rocha et al., 2021) built an ontology to support software development with distributed teams. The methodology adopted for planning and developing the ontology is the METHONTOLOGY. Based on this the first step is to define the competence issues, the second is to identify the terms of the knowledge domain and then do the identification of the classes of the ontology. The next procedure is to construct the Ontology itself, and eventually, the ontology is checked for syntactic and semantic quality.

Management Research and Practice
Volume 13, Issue 4 / December 2021

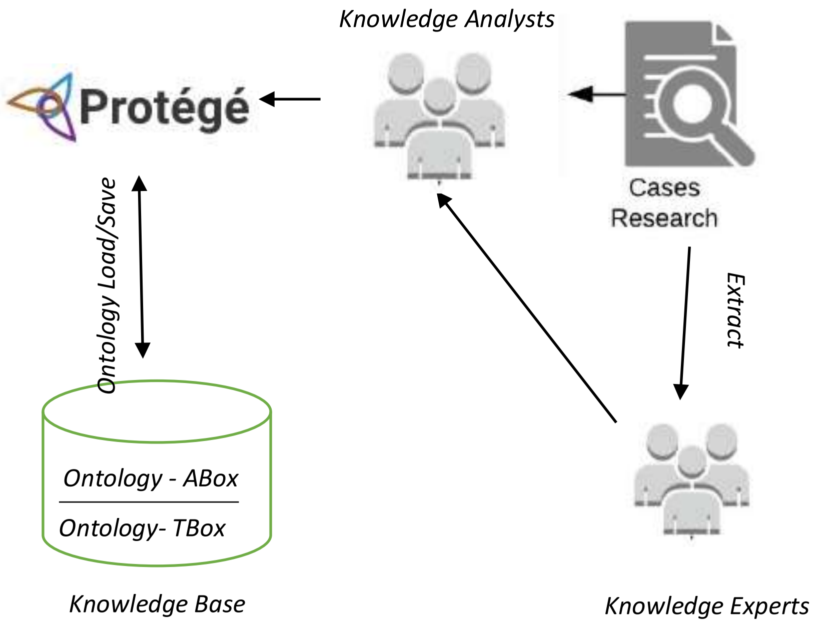

FIGURE 2 - A GENERAL VIEW OF APPROACHES USING PROTEGE.

(Yameng et al., 2021) builds OntoPLC, a semantic model of PLC programs for code exchange and software reuse. The industrial automation domain a requires highly accurate knowledge for maintenance and decision making. Thus, the ontology is constructed by defining the application first, then defining the application scenario, and subsequently retrieving the corresponding data. The ontology model is used for knowledge alignment of heterogenous platforms. A unified ontology model is built by implementing ontology matching and mapping rules. Evaluation considers clarity, extendibility, and minimal encoding bias.

4. TOOLS OF PROTÉGÉ IDE IN ONTOLOGIES DEVELOPMENT ACTIVITIES

Many research projects use Protégé for ontology editing (Musen,2015). Protégé is an open source tool with many versions available during its existence, currenly being in use version 5.5 (Protege, 2021). It contains a frame-based and an OWL-based ontology editor. Protégé OWL editor is a plug-in for the Protégé, which allows building and editing ontologies in OWL ontology language, which is a W3C (World Wide Web Consortium) recommended ontology language (OWL, 2021). Protégé is an Integrated Development Environment (IDE) using various reasoners (HermiT, Pellet, FacT++) (Pellet,2021) for ontology development to examine the consistency of that. To depict the class hierarchy of ontology and ontology navigation, are used OWLViz (OWLViz, 2021) (GraphViz,2021) and OntoGraf (OntoGraph, 2021) as Protégé plugins. OWLViz is used to see the asseted or inferred class hierarchies to compare the class of an explicit (asserted) class and a class derived through an inference (inferred class) (Figure 6, Figure 7). OntoGraf can be used to see the relationship of the ontology: classes, object properties, individuals, and equivalence (Figure 5).

Ontology editing creates elements including classes and properties, property ranges and domains, cardinalities restrictions, data types properties and individuals. This resulted in a machine readable ontology in OWL/XML format, which is an .owl file that can be saved, open or imported in the tool. Figure 3 displays a screenshot of the implementation of an ontology in Protégé. The left side of Figure 3 shows a view of the ontology including its classes and taxonomy structure. Classes with necessary and sufficient axioms are known as defined classes in OWL. It can be distinguished defined classes marked with yellow bullets and white small lines inside. On the right of Figure 3, the properties between a selected class and other classes in the created ontology are shown as well as the restrictions on these properties. A created class can be disjoint from the other classes and this axiom is editable .

The Protégé screenshot in Figure 4 depicts the properties of an ontology on the left and a selected property domain and ranges on the right. For instance, the property is highlighted on the left side of Figure 4 and the

Management Research and Practice
Volume 13, Issue 4 / December 2021

concepts `Jucator` and `Echipa` are shown as the domain and range of the property `joacaIn`, respectively. The domain is the concept described by the relation/property. One of the main differences between OWL and other object-oriented models is that properties in OWL are first class entities that are not bundled with a specific class. In OWL properties are independent entities. Properties in OWL are equivalent to binary relations in First Order Logic (FOL). They also have a number of capabilities that relations in FOL have and that can be automatically enforced by an OWL reasoner. Since OWL properties are FOL relations they are sets (of binary tuples). Thus, just as classes can have subclasses where the subclass is a subset of the superclass so properties can have sub-properties where all the tuples in the super-property are in the sub-property but not necessarily vice versa. Figure 4 gives a description of a property editing elements.

The graph representation of the ontology is presented in Figure 5. This graph is obtained with OntoGraf, a Protege plug-in for ontology visualization.

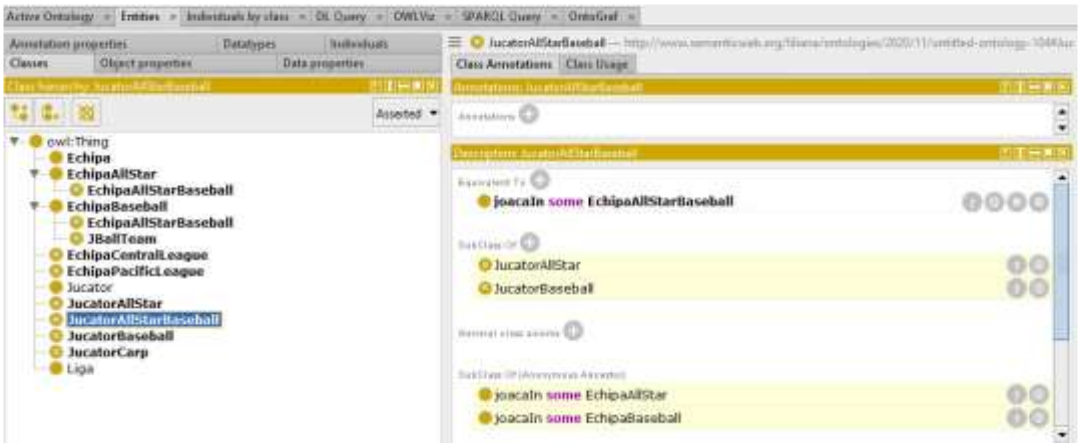

FIGURE 3 - CLASS HIERARCHY EDITING A CLASS DESCRIPTION.

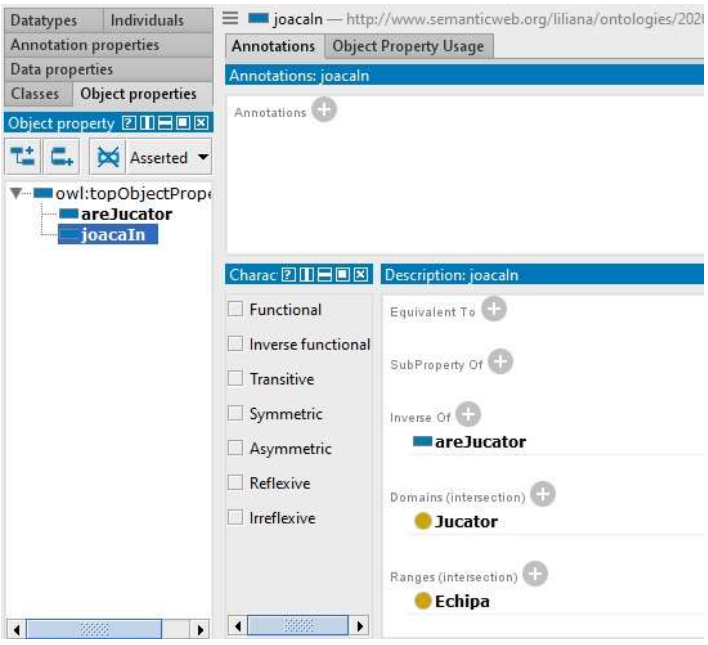

FIGURE 4 - OBJECT PROPERTIES HIERARCHY AND DESCRIPTION.

Management Research and Practice
Volume 13, Issue 4 / December 2021

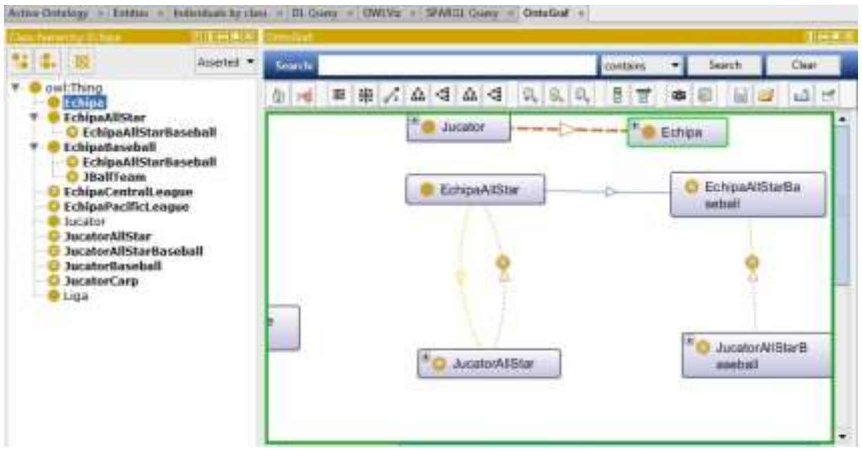

FIGURE 5 - ONTOLOGY VISUALIZATION WITH ONTOGRAF.

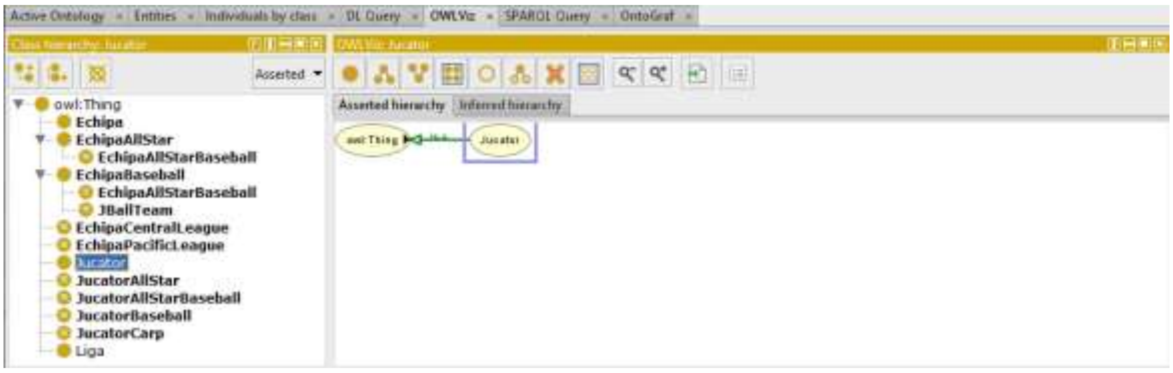

Figure 6. Ontology visualization with OWLViz – asserted hierarchy.

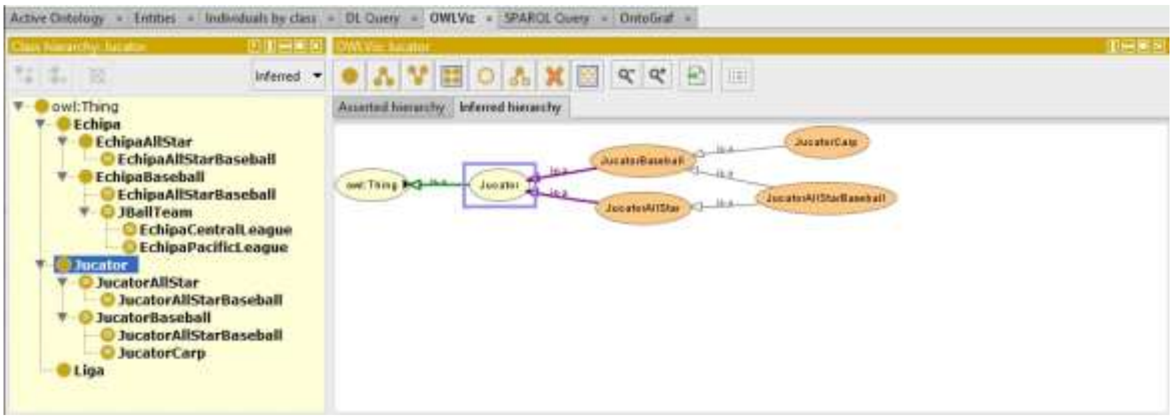

FIGURE 7 - ONTOLOGY VISUALIZATION WITH OWLVIZ – INFERRED HIERARCHY.

OWL can be used to define axioms that are necessary and sufficient for an individual to be a member of a class. The OWL reasoners can use these axioms to automatically restructure the class hierarchy in an inferred hierarchy (see Figure 7) as well as to do significant additional reasoning about individuals. If one defines axioms for a class in the `SubClassOf` field in Protégé these are necessary axioms for the class. i.e., they must be true for any individual that is a member of that class, but it may not be the case that every individual that fulfils that axiom is a member of that class. When axioms are defined in the `EquivalentTo` field in Protégé these axioms are both necessary and sufficient conditions for that class. i.e., any individual that satisfies those axioms is automatically inferred to be an instance of that class (see Figure 3). Figure 6 and Figure 7 presents ontology visualization with OWLViz, the asserted and the inferred hierarchy, respectively. Notice that anything in Protégé

Management Research and Practice
Volume 13, Issue 4 / December 2021

highlighted in orange was not defined by some manual input data, but was inferred by the reasoner based on the asserted concepts and the axioms in the ontology.

Query retrieval in Protégé can be done in two different ways, DL Query and SPARQL Query. Both are tabs in the tool user interface and plug-ins for searching a inferred ontology. Queries works only when the ontology is consistent, hence an ontology has to be checked before for consistencies. Queries are designed in order to verify and validate an ontology.

5. CONCLUSIONS

In the context of complex ecosystems ontologies are recognized strong instruments for sharing and reusing of knowledge. Also they are considered to assist the distributed software development processes, they allow the evolution of communication by setting the knowledge and information sharing in a clear and cohesive way among the people involved in a global project. The use of ontologies makes it possible to share a common understanding of the information structure between members of distributed organizations or between intelligent machines.

This study analyzed existent approaches in building ontologies and presented the capabilities of an IDE to be used in the development of an ontology. Building domain ontologies is not an easy task as it requires a lot of effort and time for domain conceptualization. The advantages of using an ontology is to reuse knowledge or when integrated into an application it allows easily access and retrieve the knowledge that can be processed by humans or machines.

Many approaches described in this study highly recommend developing new ontologies considering reuse. The reuse of existing ontologies enables to reduce time, cost and expert's knowledge required when building a new ontology from scratch. Another reason for reusing existing ontologies in the process of building a new one is that these ontologies have already been approved and tested in various applications and their reuse may yield a more accurate and reliable ontology.

An important result of the study is that the evolution of software engineering processes influences the evolution of ontology development approaches. The global and agile aspects of the development activities are among the common properties of both. The main contributions described in this article are related to finding out a variability in the development activities and their flows of control, practical illustration of processes of manual or automated classification in an ontology and how ontologies are evaluated.

ACKNOWLEDGEMENTS

This work was conducted using the Protégé resource, which is supported by grant GM10331601 from the National Institute of General Medical Sciences of the United States National Institutes of Health.

REFERENCES

Ciancarini, P., Nuzzolese, A.G., Presutt,i V., Russo, D., *SquAP-Ont: an Ontology of Software Quality Relational Factors from Financial Systems*, Semantic Web, 0(0), IOS Press, 1-14.

Colesca, S.E., Dobrica. L., Alpopi, C., 2009, *Students outcomes and perceptions in a blended learning format*, Metalurgia International, 14(8), 222-229.

De Nicola, A., Missikoff, M, Navigli, R., 2009, *A software engineering approach to ontology building*, Information Systems, 34, 258-275.

De Nicola, A., and Missikoff, M. , 2016, Methodology for Rapid Ontology Engineering, Commun. ACM, no.3, 79–86.

Dhingra, V., and Bhatia, K.K, 2015, *Development of Ontology in Laptop Domain for Knowledge Representation*, Procedia Computer Science 46, 249-256.

Management Research and Practice
Volume 13, Issue 4 / December 2021

Dobrica, L., 2021, *Knowledge in action towards better knowledge management in organizations*, in Management Research and Practice, 13(2), 26-35.

Dobrica, L., 2018, *On understanding of service ecosystems based on building a conceptual space*, Management Research and Practice, 10(4),76-88.

Dobrica, L., and Niemela, E., 2008, *A UML-based variability specification for product line architecture views* , Procs. of the Third International Conference on Software and Data Technologies (ICSOFT 2008), vol. SE, INSTICC Press, 234-239.

Dobrica, L., and Pietraru, R., 2017, *Experiencing native mobile health applications development*, The 21st International Conference on Control Systems and Computer Science (CSCS), pp. 523-528.

Dutta, B., Chatterjee, U. and Madalli, D, 2015, YAMO: Yet Another Methodology for large scale faceted Ontology construction, Journal of Knowledge Management. Vol. 19, no. 1,. 6-24.

Fernández-López, M., Gómez-Pérez, A., and Juristo, N., 1997, *METHONTOLOGY: From Ontological Art Towards Ontological Engineering*, AAAI-97 Spring Symp. Ser., vol. SS-97-06, 33–40.

GraphViz, 2021,www.graphviz.org

Gruber, T. R., 1993, *A Translation Approach to Portable Ontology Specifications*, Knowledge Acquisition, Vol. 5 No. 2, 199–220.

Gruber, T.R., 1995, Toward principles for the design of ontologies used for knowledge sharing. Int. J. Hum.-Comput. Stud. 43 (5-6), 907–928.

Gutierrez, C., and Sequeda, J. F., 2021, *Knowledge graphs*, Communications of the ACM, 64(3), 96-104.

Hitzler P., 2021, A review of the Semantic Web Field, Communications of the ACM, 64(02), 76-83.

Hoekstra, R., 2017, Ontologies & Ontology Engineering, 80 slides.

Maduray, K., Naidoo, S.,C, Pillay, K., and Fonou-Dombeu, J.V., 2020, *OntoEskom: An ontology for the supply and generation of electricity by ESKOM*, 20202 Int. Conf. On Artificial Intelligence, big data, computing and data communication systems (icABCD), IEEE.

Menzel, C. P., and Mayer, R. J., 1992, IDEF5 Ontology Description Capture Method : Concepts and Formal Foundations.

Musen, M.A., 2015, *The Protégé Project: A Look Back and a Look Forward*, A.I. Matters ACM SIG in Artificial Intelligence, 1(4).

Noy, N. F., and McGuinness, D. L., 2001, *Ontology Development 101: A guide to creating your first ontology*, Stanford University, Stanford, CA.

OWL, 2021, http://www.w3.org/2001/sw/wiki/OWL and http://www.w3.org/TR/owl2-primer/

OWLViz, 2021, http://protegewiki.stanford.edu/wiki/OWLViz.

OntoGraf,2021, http://protegewiki.stanford.edu/wiki/OntoGraf

Protege, 2021, http://protege.stanford.edu.

PELLET, 2021, http://clarkparsia.com/pellet and http://pellet.owldl.com/

Pustovalova, N., Avdenco, T., and Pustovalova, A., 2021, Using Ontologies for the development of the university educational ecosystem, 2021 IEEE 22nd International Conference of Young professionals in electron devices and materials, 568-571.

Rocha, R., Leandro, R., Silva, I., Araujo, J., Bion, D., Freitas, F., Cordeiro, D., Gomes, A., and Azevedo, R., 2021, *DKS-S: An ontology-based tool for global software development*, The 16th Iberian Conference on Information Systems and Technologies (CISTI).

Schachinger, D., Kastner, W., and Gaida S., 2016, *Ontology-based abstraction layer for smart grid interaction in building energy management systems*, 2016 IEEE International Energy Conference (ENERGYCON), IEEE.

Screiber G., Wielinga B., Akkermans, H., van de Velde, W, and Anjewieden,A., 1994., *CML: The CommonKADS Conceptual Modelling Language*, Procs 8th European Knowledge Acquisition Workshop, 1-25.

Screiber G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., van de Velde, W., and Wielinga, B., 2000, Knowledge engineering and Management, The MIT Press.

Shrivastava, S., Mathur, I., and Joshi, N., 2018, *An ontology development for university*, 2018 International Conference on Advanced Computation and Telecommunication (ICACAT), IEEE

Studer, R., Benjamins, V. R., and Fensel, D., 1998, *Knowledge engineering: Principles and methods*. Data & Knowledge Engineering, Vol. 25 No. 1, 161–197.

Sure, Y., Staab, S. and Struder, R., 2004, *On-To Knowledge Methodology*, Handb. Ontol., 117–132.

Ungkawa, U., Widyantoro, D.H., and Hendradjaya, B., 2017, *Methodology for constructing Form Ontology*, Procs. 2017 Int. Conf. On Electrical Engineering, Computer Science and Informatics (EECSI), IEEE.

Wilkinson, M., Dumontier, M., and Aalbersberg, I., 2016, *The FAIR Guiding Principles for scientific data management and stewardship*. Sci Data 3.

Yameng, A, Feiwei Q., Baiping, C., Rene, S., and Huifeng, W, 2021, *OntoPLC: Semantic Model of PLC Programs for Code Exchange and Software Reuse*, IEEE Transactions on Industrial Informatics, 17(3), 1702-1711.

Management Research and Practice
Volume 13, Issue 4 / December 2021