ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

Role of Ontologies in Software Engineering

[1]Karthick
[1]Department Of Computer Science and Engineering, Galgotias University, Yamuna Expressway Greater Noida, Uttar Pradesh
[1]karthick@Galgotiasuniversity.edu.in

Abstract: The ongoing process of semantic web technology guarantees new opportunities for analysis into software development.Nonetheless, because the semantic web's fundamental principles have a strong tradition in the area of information technology, it is sometimes difficult for engineers to neglect the range of ontology-enabled strategies to Software Engineering.The word ontology is becoming prominent in various computer science fields such as Machine learning, Informant Systems, Registry or Web Innovations.Diverging from its initial intellectual significance, the word ontology stands for a structured explicit configuration of a shared categorization in the sense of computer engineering.Software Engineering (SE) is an area where conceptualization plays a significant role, e.g. in the initial stages of application development, in the implication, use and reprocess of software tools, as the premise for the inclusion. So ontologies are probably going to conquer the Software Engineering area.The use of ontologies in each step of software engineering provides a strategic advantage across conventional approach, leading to greater exchange and retrieval of knowledge.

Keywords: Advantages of Ontology, Lifecycle of Ontology, Ontologies, Software Engineering.

INTRODUCTION

The "software engineering and knowledge engineering" cultures share a number of commonalities.Although software engineering work has constantly sought a greater degree of subjectivity and has emphasized software designing over the last century, the information academic community has been keen to encourage many modelling strategies to recognize the semantic vision.The correlation becomes more apparent with the introduction of cloud-based software, and in particular web providers. All groups, nevertheless, exist largely in their very own environments.The use of the "Software Engineering Environment (SEEs)" can enhance software performance and efficiency[1].Ontologies are an effective way of describing and arranging subject matter expertise in a Domain oriented software engineering environments.Creating ontologies, though, includes defining the definitions and connections that occur in the field, in addition to their meanings, attributes and restrictions.Rapid growth of communications and mobile technology, constant requests for new technologies and an enormous number of web users are among the reasons why more software is continually being required.It obviously involves effective strategies for software development that will be able to react properly to the requirements for which the software is constructed, and yet enable for greater levels of software technicians efficiency.

However, the practice shows that both points of view are still suffering from severe issues[2].Although software is a technological classification developed by using computer hardware to automate repetitive functions, it is also a sociological phenomena used in virtually any aspect of society today.In addition, software is a database of information whereby information is due entirely to the context of the program, and not software as an object.Therefore, there is a need to be able to exchange and modularize (implementation) expertise contained in operating system with understanding of all appropriate software-related elements (e.g. database knowledge, new demands, initiatives, and situations in which individuals use communicate with software) in order to put software to more proficient level.Such information exchange and governance involves the use of clear information description, because computers have to be able to understand information. This is why ontologies have been accepted by the software engineering community as a successful way of addressing new software engineering challenges.

Ontology

The term "ontology" has been commonly used in many areas of computer science such as Machine intelligence, Manager Systems, Information technology, Server or Web Technology over the last few years. Different techniques and methods have been written and distributed for their interpretation and usage. Currently, the ontology method has been regarded and discussed in parts of the Software Engineering (SE) and Enterprise applications groups, primarily because of its similarity to the (theoretical and enterprise) modelling areas, the UML framework and the areas of duplication and incorporation of software components[3].Ontology simply implies the "science of being"-more correctly: it interacts with the implications and circumstances of the being. In other terms, scholars are wondering what "being" implies, what are the preconditions and limits of "being," its source and possible perspectives.Usually people accept what they can examine, think, contact or gain from such stimuli through rationale as well as other psychological perception without queries as "being."Yet opinions and beliefs about the same "being thing" may differ from one individual to another.Researchers, unlike most theorists, adopt a naïve, "practical" method[4].

Ontology can't even express "what's" and at most what people think they know about certain region of the world and what they can interact through language which are available.Such an interaction occurs in Informatics at many levels: between users and developers of a proposed or current computer system while evaluating its specifications and (re)designing or integrating the system features, among users and programs or the modules while operating via a user interface or between modules of systems or electronic tools while operating with one another.Ontologies provide a standard, mutual understanding of a field that can be exchanged between individuals and software frameworks.Ontologies have the function of collecting subject matter expertise in a technical way and offering a widely accepted understanding of a field.The prevalent ontology language, which defines the significance of aspects and the relationships, is typically organized in a categorization and includes primitive modelling like classes, relationships, functionalities, and theorems.Ontology is a specific conceptual definition of a common conceptual framework.

ONTOLOGY IN SOFTWARE ENGINEERING

Software Engineering is the use of a structured, controlled, quantifiable method to software engineering, construction and maintenance.There has also been a never-ending quest to increase the level of formalism via modelling and greater-level programming disciplines to deal with the computational complexity of the software.Though, many problems were solved only partly including manufacture of components, structure, verification, data and assimilation of applications, software testing and performance[5]. These basic issues are the incentive for new methods that affect every element of software engineering.Modularization, production, reuses and convergence of software components and structures has been one of the main Software engineering issues from its inception. The more extensive and computerised such tasks become, the more significant is the concept and use of ontologies as a theoretical premise for such components.When systems or modules are to share information, this will occur more on the design than on the stage of execution. A great potential for developing, applying, distributing or using ontologies is emerging in the age of uniting software environments.SE is a technical and science domain of its own with its own framework and terms. The task of constructing one or several ontology for this area is a required and difficult task because it is a very new domain.Ontology development and software design have its own, overlapping, interconnected processes that have something in particular but also vary in their objectives roles, and time frames.The advancement of ontology is much more brief-term driven, and has a broader spectrum.It is usually connected to several software systems and operated by groups and companies covering programs.In a first effort, ontology could be regarded in the software development process as a unique element. However, if it includes the entire application realm, it is more likely to influence understanding to be used in several elements of the structure to be established.

ONTOLOGY IN SOFTWARE ENGINEERING LIFE CYCLE

*Analysis and Design:*During this phase of Software Engineering life cycle, several key fields of implementations were established where ontology could play an important role.Firstly, requirement technicians may benefit from conceptual frameworks in view of describing information and supporting processes.Second, throughout layout the recycle of components is selected as a prospective implementation region.

*Requirement Engineering:*Ontology is used for both to define specification files of criteria and to officially reflect understanding of requirements.Natural language is mostly used to define specifications, i.e. in the type of use cases[6].Nonetheless, the use of standard vocabulary or of structured specification languages is necessary.In addition, ontologies could be used to support control and quality controls of specifications that enable software engineers recognize the relationships and interactions between different software artworks.

*Component Reuse:*Generally the quest for usable elements occurs after the design process, when the operational specifications are established. Because most databases are restricted to a search based on a single lexical word, they suffer from low specificity and poor recall.Ontologies can tend to define the features of modules using an abstraction of information interpretation that makes for more simple and efficient queries. The various steps of ontology process in software engineering are shown below in figure 1, Ontology process in Software Engineering.

*Implementation:*A crucial step in the software development process moves from research and specification to execution.The question is raised how to exploit ontologies to close the gap among design and construction.Several areas of concern are concurrent machine simulation of ontology words and applying ontology to operating time."Ontology as a domain model" Because a domain model is originally uncertain and constantly changes, if not needed, a single abstraction and differentiation of considerations is deemed practicable.Incorporating with Computer Modelling Languages, Many alternatives are available for incorporating languages and ontology systems based on MDA knowledge representations[7]. While some find the UML to be the language of ontology representation, others use the UML as a model language for ontology creation.Sequencing a domain model to program should be computerized so that other elements and programs can make vibrant use.

*Coding Support:*In Object Oriented coding, it is the norm to detach configuration requirements (such as Java Functionality, conceptual class) from their application to make demanding apps autonomous of inner modifications.The developers have to deal with huge range of APIs, so recording such APIs has become an important consideration.

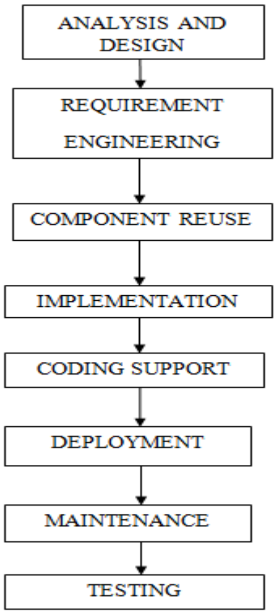

Fig. 1: Ontology Process in Software Engineering

*Deployment:*Problems such as module interconnections or legal restrictions make managing virtualization structures a complicated job. In this case ontologies provide a framework for gathering information on the field of problems.The "business logic" is built directly in computer languages for most software systems. Therefore improvements to a software system's core functionality include modifications to the data files.Business rule algorithms are a potential way to solve this issue. Separating business logic and production logic is the fundamental concept.

Maintenance: Ontologies help direct programmers electronic communication with error feedback and the affected regions in the source code. The culture, their connections and quality are core principles.Numerous types of relevant information occur in software maintenance business processes without a specific linkage.This is troublesome, as a comprehensive perspective would prevent repetitive research and improve analytical thinking.For instance, a bug solving mechanism usually includes the exploration and investigating of a bug, general discourse within a programmer unit and eventually alters to the software that addresses the bug.

*Testing:*Ontologies may help create simple test cases because they represent subject matter expertise in a form that can be interpreted by machines.One similar example for this would be the restrictions on cardinality.Because these limitations identify limitations on the connection of some classes, Ontologies can be used for examining equivalence classes.Ontologies can actually create basic test cases because they encrypt domain knowledge in a layout that can be processed by device.

ONTOLOGY LIFE CYCLE

Ontology is a unique piece of "software" that has its own life-cycle mainly somewhat long term.Such a life cycle is interwoven with the life cycles of software product development projects which are attached by the very same application framework to ontology[8].All life cycles are taken up by various duties "the software engineer and the ontology engineer."Generally, ontology design does not adopt a linear schedule with successive stages, intermediary outcomes benchmarks, etc- while specific topics (ontology elements) could be established in this manner.Incremental approach seems suitable, beginning with a kernel ontology, which is slowly expanded by "ontology increments," i.e. limited formalizations of unique sub domains.

The various phases of Ontology life cycle are:

*Analysis:*In scenario of constructing a new ontology, the first stage provides to begin the procedure of ontology advancement.This defines the nature of discussion, aims and intentions, and establishes potential sources of information.Interfaces and links to other current, related ontologies or elements (previous, ancestor, contemporary, neighbour, conflicting, opposing types) are examined[9]. The function of use cases is performed by established or potential implementations, i.e. programs which (could) use the ontology now or in the near.A general ontology strategic plan is described, along with an approach for its further enhancement and assimilation.In the scenario of a reworking a current ontology, research continues and develops on the current concepts. The various steps of ontology life cycle is shown below in Fig. 2 Ontology Life Cycle

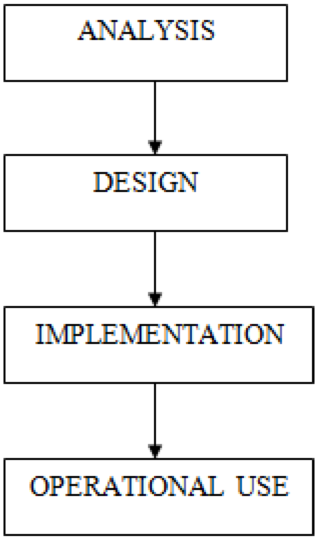

Fig. 2: Ontology Life Cycle

*Design:*In this step, framework and hierarchy, potential sub-ontologies, frameworks and commitment guidelines are defined. The ontology field is conceptualized, the dictionary is (additionally) packed, expanded, and increased, descriptions and cross-references are provided.Mappings and interpretations are given into "dialects" such as E / R- or UML drawings or structured ontology languages. Concepts are tested for comprehensiveness and accuracy.

*Implementation:*The ontology is converted into a specific ontology or web application ("e.g., categories DAML+OIL, OWL, XML+RDF, DL, Prolog, Java") as needed. The outcomes of the sub-development cycles (in terms of subontology) are incorporated.The ontology concepts for (prospective) customers are released.

*Operational use:*The process of unification is done with ancestral and neighbouring ontologies.Integration components are searched for intersecting sections, inaccuracies, vagueness, etc. Suggestions is collected and examined from app programs, organizations, customers, etc.A review process might be introduced if justified by additional requirements and the organizational requirements.

BENEFITS OF ONTOLOGY IN SOFTWARE ENGINEERING

Because designing ontologies is a tiresome and expensive challenge, it is always essential to show the benefits that can be gained by implementing ontologies in software development.It is underscored by the assumption that many of the ontology's structured frameworks are in position for many years, despite software developers witnessing rapid adoption.So obviously the present emergence of reasoning-based formalisms is a critical component in the "semantic web" attempt[10].In a way, the significance of optimization here can be contrasted with the graphical simulation condition in Software development before Unified Modelling Language.Some other important element is that ontologies are flexible.Furthermore, the stability makes it easy to broaden current ontologies, thus encouraging the reuse of original works. The "web "- focus of present ontology methods further promotes this.Since software systems are also becoming progressively web-enabled, and therefore have to deal with information from diverse sources which might not be understood at the point of enhancement, software developers are searching for innovations that can aid in such scenario. The network also makes knowledge-sharing simpler.With respect to even more software engineering-specific benefits, ontologies render domain designs citizens of first order. When domain designs clearly drive the core of any software platform, their significance in new software design procedures reduces after the evaluation stage.The main function of ontologies is the structured definitions of a field by nature and therefore promotes a wider use across the life cycle of Software Engineering.Ontologies are a powerful method for moving information from task to task within a given software field as well as from one project to the next development process.Ontologies may become an appealing framework for software engineering that provides for closer integration, stronger integrated designs, more reusable elements and fewer application development expenses.

CONCLUSION

The use of ontologies for multiple objects in software is likely one of the regions that has drawn much exposure so far.Domain and top-level ontologies, conceptual frameworks for information, raw data, bugs, ontology-based designs, model implementations, specifications, and layout trends are really just a few instances used for critical software engineering activities, like introducing more semiconductor to the objects, enhancing tractability connections, design reliability verification, model transition generation, and software measures.So far, creation of ontology-driven technology has concentrated about the use of ontologies in theoretical and domain analysis for field interpretation. This seems to originate from the belief that ontologies and designs of software have different motives, even conflicting ones.Such advantages are a fundamental part of a greater awareness of ontologies in Software Engineering. A secret to supporting ontology advantages is a greater reuse of ontological awareness throughout the lifespan of Software development.The greater the ontology is being used, the more database understanding in ontology is available and the less effort will need to be spent on introducing information to ontology. Subsequently, software enhancement will take far less time and debugging effort because more understanding is used again. The paper gives a comprehensive overview on the role of ontologies in Software development, the various steps of ontology in software engineering, the various stages of ontology life cycle followed by its advantages.

REFERENCES

[1] D. Strmečki, I. Magdalenić, and D. Kermek, “An overview on the use of ontologies in software engineering,” *J. Comput. Sci.*, vol. 12, no. 12, pp. 597–610, 2016.

[2] A. J. Wiebe and C. W. Chan, “Ontology driven software engineering,” in *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering: Vision for a Greener Future, CCECE 2012*, 2012.

[3] D. Jacquette, *Ontology*. 2014.

[4] R. Arp, B. Smith, A. D. Spear, R. Arp, B. Smith, and A. D. Spear, "What Is an Ontology?," in *Building Ontologies With Basic Formal Ontology*, 2016.

[5] M. I. Marwat, S. Jan, M. A. Shah, and S. Z. A. Shah, "Towards optimization of software engineering ontologies," in *2014 World Congress on Computer Applications and Information Systems, WCCAIS 2014*, 2014.

[6] D. Dermeval *et al.*, "Applications of ontologies in requirements engineering: a systematic review of the literature," *Requir. Eng.*, 2016.

[7] R. Lourdusamy and J. M. Florrence, "Methods, approaches, principles, guidelines and applications on multilingual ontologies: A SURVEY," *ICTACT J. Soft Comput.*, vol. 07, no. 01, pp. 1350–1358, 2016.

[8] C. Gonzalez-Perez, "How ontologies can help in software engineering," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017, vol. 10223 LNCS, pp. 26–44.

[9] S. De Cesare, F. Gailly, G. Holland, M. Lycett, and C. Partridge, "Ontology-driven software engineering 2010," in *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion, SPLASH '10*, 2010, pp. 279–280.

[10] Q. M. Ilyas, "Ontology augmented software engineering," in *Software Development Techniques for Constructive Information Systems Design*, 2013, pp. 406–413.

[11] Vishal Jain and Dr. S. V. A. V. Prasad, "Mapping between RDBMS and Ontology: A Review", International Journal of Scientific & Technology Research (IJSTR), France, Vol. 3, No. 11, November, 2014 having ISSN No. 2277-8616.

[12] Vishal Jain and Dr. S. V. A. V. Prasad, "Mining in Ontology With Multi Agent System in Semantic Web : A Novel Approach", The International Journal of Multimedia & Its Applications (IJMA) Vol.6, No.5, October 2014, page no. 45 to 54 having ISSN No. 0975-5578.

[13] Vishal Jain, "A Brief Overview on Information Retrieval in Semantic Web", International Journal of Computer Application, RS Publication, Issue 4, Volume 2 (March - April 2014), page no. 86 to 91, having ISSN No. 2250-1797.

[14] V.M. Prabhakaran, Prof S.Balamurgan ,A.Brindha ,S.Gayathri ,Dr.GokulKrubaShanker,Duruvakkumar V.S, "NGCC: Certain Investigations on Next Generation 2020 Cloud Computing-Issues, Challenges and Open Problems," Australian Journal of Basic and Applied Sciences (2015)

[15] V.M.Prabhakaran, Prof.S.Balamurugan, S.Charanyaa, "Data Flow Modelling for Effective Protection of Electronic Health Records (EHRs) in Cloud", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 1, January 2015

[16] R. Santhya, S. Latha, S. Balamurugan and S. Charanyaa, "Further investigations on strategies developed for efficient discovery of matching dependencies" International Journal of Innovative Research in Science, Engineering and Technology Vol. 4, Issue 1, January 2015