Community-oriented Software Engineering Ontology Evolution

P. Wongthongtham, N. Kasisopha and S. Komchaliaw
DEBII, Curtin University, Australia
p.wongthongtham@cbs.curtin.edu.au; natsuda.kasisopha@postgrad.curtin.edu.au;
maxtor_p@hotmail.com

Abstract

Software Engineering Ontology (SE Ontology) defines common shareable software engineering knowledge and typically provides software engineering concepts: what the concepts are, how they are related, and why they are related. These concepts facilitate common understanding of software engineering knowledge across multiple international software development sites. The SE Ontology is in machine understandable form to facilitate meaningful communication for remote social members. These social members use the SE Ontology but are not involved in the development process. Most existing ontologies including the SE Ontology are designed by individuals or small group of experts, not actual ontology users nor various groups of experts. It is effective if the ontology users can contribute in the process of creating and maintaining the ontologies they use. Social networking is becoming more prevalent enabling people to engage in remote collaboration to form goal-directed social networks. In this paper, we propose a social network based approach for ontology evolution for the SE Ontology. We analyze ontology evolution of the SE Ontology and propose the social network based approach for making ontology evolution more responsive to users' needs.

1. Introduction

Software Engineering Ontology (SE Ontology) [1] defines common shareable software engineering knowledge and typically provides software engineering concepts: what the concepts are, how they are related, and why they are related. These concepts facilitate common understanding of software engineering knowledge across multiple international software development sites. When this generic ontology is specialized to a particular project and populated with instances which reflect the project information it provides this common understanding of project information to all the distributed members of an international development team in a multi-site software development environment. The SE Ontology is in machine understandable form to facilitate meaningful communication for remote social members. These social members use the SE Ontology but are not involved in the development process. Most existing ontologies including the SE Ontology are designed by individuals or small group of experts, not actual ontology users nor various groups of experts. It is effective if the ontology users can contribute in the process of creating and maintaining the ontologies they use.

Social networking is becoming more prevalent with current technologies such as HTTP, RSS/Atom, and Scripting. Social bookmarking websites, blogs, Wikis are excellent examples of using Web-based technology that enables people to engage in remote collaboration to form goal-directed social networks. A direct benefit of social networking is the formation of a virtual community from which considerable value can be derived based on certain social network properties. In this paper, we propose a social network based approach for ontology evolution for the SE Ontology.

The paper is structured as follows. First we analyze ontology evolution of the SE Ontology. Then we propose the social network based approach for making ontology evolution more responsive to users' needs. Next we review literatures to prove our unique approach. Finally we conclude the work and overlook future works.

2. Software engineering ontology evolution

The SE Ontology represents the knowledge by structuring concepts, their relationships and their constraints, i.e. it is an abstract level of representation. Project data over a period of time will be populated as instances in the deployment stage. Precisely stated, the SE Ontology captures the generic software engineering concepts as well as the specific software engineering project information / data / agreements. The key ingredients that make up the SE Ontology are a vocabulary of basic software engineering terms and a precise specification of what those terms mean. The SE Ontology is populated with specific instances for a particular project for the corresponding software engineering concepts. These instances contain the actual project data being queried in the knowledge based applications. Thus the SE Ontology includes the set of actual project data (i.e. instances of the concepts) and assertions that the instances are related to each other according to the specific relations between the concepts.

Ontology evolution is defined as “the timely adaptation of an ontology to the arisen changes and the consistent propagation of these changes to dependent artifacts.” [2]. Evolution is in two senses i.e. (i) evolution in generic software engineering concepts (concepts level) which reflects new kinds of software, application, or system being developed on broader and different understandings (ii) evolution in specific project data (instances level) which reflects project development, changes in the software requirements or in the design process in order to incorporate additional functionality to software, application, or system or to allow incremental improvement and the like.

For example the concept ‘Use Case’ defined in the SE Ontology evolves as more software engineering projects use it in slightly different ways based on specific project requirements. In this example, Use Case (A, a) (i.e. Use Case “Order Processing” for project *a*) could evolve within different projects. Based on the SE ontology, a Use Case is formally defined by the set of primary attributes – x_1(actor), x_2(goal), x_3(scenario) and x_4(trigger). However, when dealing with various complex e-Business requirements, these four attributes sometimes might not be sufficient. For instance, two more local attributes, x_5(time) and x_6(quantity) should be further defined for B2B e-Business projects. The introduction of x_5 and x_6 has given rise to the concept Use Case (A, b). In a similar vein, for project *c* the secondary attribute x_7(connectivity) led to the formation of the concept Use Case (A, c) for an e-Business/ERP integration project. Attribute x_4(trigger) is not included in Use Case (A, c). This is because the integration between order processing and ERP system is not directly triggered by an actor (i.e. a customer) but by internal business logic. From this example, one can see that ontology evolution plays a key role during the software development for different projects.

3. Community-oriented approach for SE Ontology evolution

The conceptual framework of community-oriented software engineering ontology evolution is shown in Figure 1. It is grounded in the notion of several key architectural components and data elements.

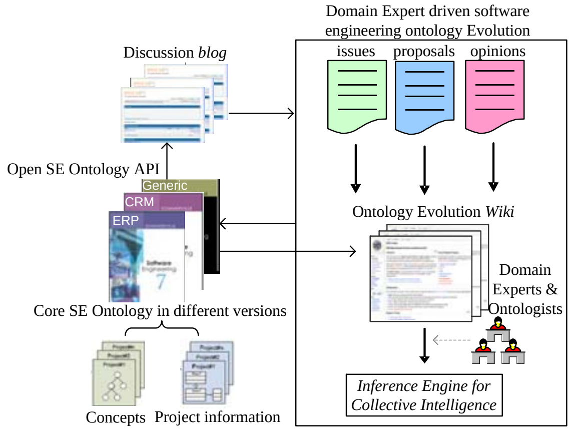
Domain Expert driven software engineering ontology Evolution
issues
proposals
opinions
Discussion *blog*
Open SE Ontology API
Generic
CRM
ERP
Ontology Evolution *Wiki*
Domain Experts & Ontologists
Core SE Ontology in different versions
Inference Engine for Collective Intelligence
Concepts
Project information

Figure 1. System overall architecture

We use community-oriented approach in two ways i.e. using blog and Wiki. Blog is something more personal and people are more willing to emerge in discussion while wiki is something that people prefer to use when ideas are more concrete.

Once the ontology evolution process is completed and the community users have reached agreements on what changes are to be made, the changed concepts are merged with the existing SE ontology which could make use of another version of the SE Ontology and updated in an inference engine for collective intelligence. It can be seen that our ontology evolution approach considers both users opinions and expert decisions.

3.1. Discussion blog

The discussion blog enables developers to share, discuss, comment, tag issues and problems in an informal and light-weight manner. The blog appears to be similar to many existing blogs e.g. beehive that have been used for remote communications. However, our discussion blog is different because we integrate the SE Ontology into it through an open SE Ontology API. In the discussions, community users can tag the particular discussion as it is issues or opinions.

In a prototype system, we use Apache Tomcat for our server to run Webprotege [3], MySQL, and PHP. Wordpress [4] and its plug-ins are used for creating discussion blog. It is also linked to the SE Ontology through Webprotege.

3.2. Ontology evolution Wiki

The domain expert driven software engineering ontology evolution is placed around an Ontology Evolution Wiki that can be collaboratively edited and annotated by a group of domain experts. The ontology evolution wiki is used by domain experts to express understanding of certain concepts and interpret them in ways different from current ones defined in the SE ontology. In this way, people can see how the meaning of software engineering terms has emerged through discussion in this community-oriented approach. The collective knowledge within this Wiki can then be acquired through various text and data mining techniques. In the domain expert driven software engineering ontology evolution, three types of important architectural data elements are: issues, proposals, and different opinions. Project issues represent unresolved queries. Proposals represent the latest/different understanding on project-related information. Different opinions include all requests that do not comply with the semantics, relations, and concepts defined in the SE ontology. These different opinions form the basis for ontology evolution facilitated by Wiki and RSS/Atom. Some requests that do not match the existing SE ontology become "different opinions". These different opinions are stored in the database and represented in an ontology modification page in the ontology evolution wiki as shown in Figure 2.

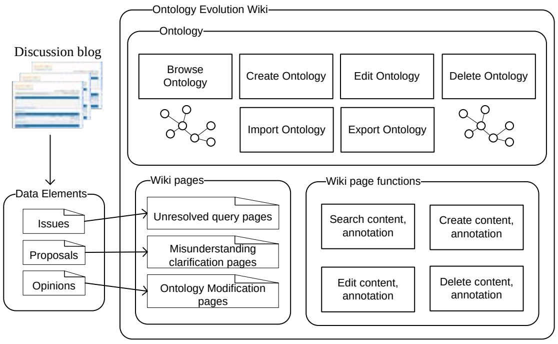

Figure 2. Ontology evolution Wiki architecture

We use an open-source system to create the ontology evolution wiki i.e. semantic media wiki [5] with integration of the SE Ontology. The complete sets of revision histories stored in a Wiki server provide a preliminary tracking mechanism that facilitates "the quality assessment" extremely essential to the project solution recommendation for ontology evolution. For example, [6] has suggested that Wiki articles' qualities and contributions can be effectively measured using these revision editing data. However, [6] focus on the quality of each article, whereas our work focuses more on the quality of each concept/term edited/defined on the Wiki. This requires not only distinct quality measurement models but also effective information extraction techniques that can capture important concepts from edited Wiki articles. [7] have made considerable progress in mining semi-structured information and limited forms of partial structured natural language which can be further studied and leveraged by our system in order to facilitate the "extraction and mining" process. However, to our knowledge, limited efforts have been made to assess the quality of the concepts defined on the Wiki platform within a social network. Once the quality assessment, data mining and information extraction are done domain experts and ontology engineers will collaboratively discuss and decide concepts that need to be updated.

4. Related works

Some work has been carried out in the areas of ontology evolution and social networking. So far no one has used a social network based approach for the purpose of ontology evolution of the SE Ontology.

4.1. Ontology evolution

Blundell and Pettifer [8] developed a plug-in in Protégé [9]. They discussed an idea of ontology evolution in graph visualization and surmised 12 visualization requirements that are applicable for ontology evolution. Noy et al. [10] developed 2 Protégé plug-ins i.e. Change management plug-in and PROMPT plug-in. They also introduced the Change and Annotations Ontology (CHAO). All changes were stored in CHAO instances and were presented in the change management plug-in. PROMPT plug-in was used to examine a list of users who made changes and to accept or reject changes. Plessers et al. [11] developed a framework which consists of evolution on request part and evolution on respond part. They also developed 2 Protégé plug-ins i.e. version log generator plug-in and change detection plug-in. Liang et al. [12] discussed facilities providing in Protégé in regard of ontology evolution. The above literatures are attached to Protégé which restricts to only ontology editor system.

4.2. Social networking

In this section, we review social networking systems in regard of ontology evolution.

4.2.1. Wiki. Many research centers and organizations had proposed and implemented a wiki to form a community-driven ontology evolution system. The semantic wiki such as Ikewiki [13], Semantic Media Wiki [5], OntoWiki [14], and Kaukolu wiki [15] were purposely implemented to facilitate users with Semantic wiki characters and features to support the ontology evolution in different perspectives.

4.2.2. Floksonomy. Gendarmi and Lanubile introduced floksonomy to indicate the act of collaboratively tag resources within communities [16]. Their vision was to develop a community which allows users to participate including add, delete, and modify the ontology. Open voting system was used in the ontology evolution.

4.2.3. Portal. Siorpaes [17] proposed an idea of using portals to involve users in the ontology evolution process.

5. Conclusion and future works

We have proposed the community-oriented approach for ontology evolution for the SE Ontology. We analyzed ontology evolution of the SE Ontology and proposed the approach for making ontology evolution more responsive to users' needs. In our future work, we will integrate software agent with the community-oriented software engineering ontology evolution. We will aim to develop a recommender approach to provide active support and recommendations to remote software engineers. We will create an open knowledge platform with active support through recommender agents situated in the foreground of the SE Ontology that interact and mediate between the ontology and human agents.

6. References

[1] Wongthongtham, P., et al., *Development of a Software Engineering Ontology for Multi-site Software Development.* IEEE Transactions on Knowledge and Data Engineering, 2008.

[2] Stojanovic, L., *Methods and Tools for Ontology Evolution*. 2004, University of Karlsruhe.

[3] *WebProtege*. 2009 [cited 2009 20 April]; Available from: http://protegewiki.stanford.edu/index.php/WebProtege.

[4] *WordPress*. 2009 [cited 2009 20 April]; Available from: http://en.wikipedia.org/wiki/Wordpress.

[5] Krötzsch, M., D. Vrandecic, and M. Völkel. *Semantic mediawiki*. in *Proc. 5th International Semantic Web Conference (ISWC06)*: Springer.

[6] Lim, E.P., et al. *Measuring Qualities of Articles Contributed by Online Communities.* in *IEEE/WIC/ACM Conference on Web Intelligence*. 2006.

[7] Tan, H., et al., *Tree model guided candidate generation for mining frequent subtrees from XML documents.* ACM Transactions of Knowledge Discovery in Data, 2007.

[8] Blundell, B. and S. Pettifer. *Graph Visualization to Aid Ontology Evolution in Protégé*. in *7th International Protégé Conference*. 2004. Bethesda, Maryland, USA.

[9] *Protege*. 2009 [cited 2009 20 April]; Available from: http://protege.stanford.edu/.

[10] Noy, N., F., et al. *A Framework for Ontology Evolution in Collaborative Environments*. in *5th International Semantic Web Conference*. 2006. Athens, USA.

[11] Plessers, P., T. Troyer, and S. Casteleyn, *Understanding ontology evolution: A change detection approach.* ScienceDirect, 2007: p. 39 - 49.

[12] Liang, Y., H. Alani, and N. Shadbolt, *Ontology Change Management in Protégé*, in *AKT DTA Colloquium*. 2005: Milton Keynes, United Kingdom.

[13] Schaffert, S., R. Westenthaler, and A. Gruber. *IkeWiki: A user-friendly semantic wiki*. in *Demos and Posters of the 3rd European Semantic Web Conference, ESWC*. 2006.

[14] Hepp, M., D. Bachlechner, and K. Siorpaes. *OntoWiki: community-driven ontology engineering and ontology usage based on Wikis*. in *Proceedings of the 2006 international symposium on Wikis*. 2006. Odense, Denmark: ACM.

[15] Malte, K. *Kaukolu: Hub of the Semantic Corporate Intranet*. in *Proceedings of the First Workshop on Semantic Wikis -- From Wiki To Semantics*. 2006: ESWC2006.

[16] Gendarmi, D. and L. Filippo, *Community-Driven Ontology Evolution Based on Folksonomies*, in *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*. 2006, Springer. p. 181-188.

[17] Siorpaes, K., *Lightweight Community-Driven Ontology Evolution*, in *6th International and 2nd Asian Semantic Web Conference*. 2007: Busan, Korea.