Ontology Learning from Software Requirements Specification (SRS)

Muhammad Ismail(✉)

Department of Computer Science and Informatics, School of Engineering, Jönköping University, Jönköping, Sweden
muhammad.ismail@ju.com
http://www.ju.se

Abstract. Learning ontologies from software requirements specifications with individuals and relations between individuals to represent detailed information, such as input, condition and expected result of a requirement, is a difficult task. System specification ontologies (SSOs) can be developed from software requirement specifications to represent requirements and can be used to automate some time-consuming activities in software development processes. However, manually developing SSOs to represent requirements and domain knowledge of a software system is a time-consuming and a challenging task. The focus of this PhD is how to create ontologies semi-automatically from SRS. We will develop a framework that can be a possible solution to create semi-automatically ontologies from SRS. The developed framework will mainly be evaluated by using the constructed ontologies in the software testing process and automating a part of it. i.e. test case generation.

Keywords: Ontology · Ontology learning · Software requirements specification · Software development process

1 Problem Description

The process of software development includes many activities, such as requirements analysis, requirements validation and software verification, that are time consuming if not automated. One promising way to automate such time-consuming activities is to use ontologies, because ontologies are machine processable models, support reasoning that can be used to assert consistency of models, and can be used in all phases of the software development process [1–3].

System specification ontology (SSO) is an ontology that captures domain knowledge and knowledge about the software system that is being developed. As software requirements specifications are used in all phases, an SSO can be created from software requirements specifications, which can be used to support different activities in the software development process. For example, such SSO represents domain knowledge in machine processable format that could help in test case generation to support software testing [4,5].

P. Ciancarini et al. (Eds.): EKAW 2016 Satellite Events, LNAI 10180, pp. 251–255, 2017.
DOI: 10.1007/978-3-319-58694-6_39

Ontologies can be developed manually, semi-automatically or automatically. Manual development of ontologies is a labor-intensive, time-consuming and expensive [6,7]. Ontology and domain experts process all information manually and should be participating throughout the manual ontology development process, therefore expensive. It is beneficial to create ontologies semi-automatically or automatically to save time and resources.

Ontology learning is a term given for the semi-automatic or automatic construction of ontologies. Ontology learning is a process of extracting terms, identifying concepts and instances[1], relations, and axioms from data sources to create and maintain an ontology [8].

There are existing ontology learning systems, such as OntoGen [9], Text2Onto [10] and CRCTOL [11], which have implemented different methods and algorithms for ontology learning. However, they focus on extracting concepts and relations between concepts, but can not extract individual and relations between individuals from software requirements specifications.

Software requirements specifications have detailed information about requirements such as inputs, conditions, actions and expected results. In software development, this detailed information about the requirements is required to implement the functionality and automate the processes. This Detailed information about the requirements can be specified by creating individuals and defining relations between individuals in an ontology. Individuals and relations between individuals are important for reasoning, inference rules and processing in applications for practical use of ontologies [4]. Existing ontology learning methods and systems need to be improved for creating ontologies from software requirements specifications to represent requirements and domain knowledge of a software system.

To summarize, it seems that the semi-automatic creation of ontologies from software requirements specifications is a challenging problem. The focus of this PhD project is how to generate ontologies semi-automatically from software requirements specifications to use in software development processes. In practice, manual ontology development is not feasible for the software industry because it requires time, resources and expertise which will not save time and cost in the software development processes.

2 State of the Art

During the last decades, several systems and tools have been developed for ontology learning. These systems and tools work with different types of data and information resources. Several systems, such as [9–12] have been developed for ontology learning from text. The OntoGen system [9] is a semi-automatic and data-driven ontology editor for creating ontologies. The system uses machine learning and text mining algorithms, and suggests concepts and relations to the

[1] Individuals and instances are same in our context. In OWL language, individuals are known as instances as well.

users. The users of the OntoGen system have full control in the ontology construction process by accepting or rejecting the system suggestions. The users can manually adjust the properties of the suggestions for the ontology. The approach is highly interactive and the users need to know the process of ontology development and knowledge of the domain where developed ontologies supposed to use.

The Text2Onto [10] is an ontology learning framework which combines machine learning and natural language processing (NLP) approaches to learn an ontology. The idea is to get better knowledge extraction results by using several different approaches and then combining the results. The OntoExtractor [13] is an ontology learning framework for extracting seed concepts and relations from natural language text to overcome problems related to ontology refinement, axiom extraction and utilizing core concepts. The CRCTOL [11] is a domain ontology learning system that used statistical and lexico-syntactic methods to extract key concepts from a document collection. Most of existing systems and tools use statistical methods to extract terms and concepts from text and are dependent on size of text, which can affect the performance of the methods for extracting terms and concepts for creating an ontology [14]. Moreover, the existing methods extract only salient concept and can not extract specific technical concepts contained in technical documents [15]. In our case, software requirements specifications are the technical documents that will be used to learn ontologies. It is important to extract all concepts including individuals of concepts that can be used to represent information for the practical use of ontologies.

3 Aim and Objectives

The overall aim of the work is to create ontologies semi-automatically from the software requirements specifications. To reach this aim, a number of objectives have been specified:

O1. *To understand the manual process of ontology development for software requirements specifications.*
O2. *Evaluate existing ontology learning methods, to determine which methods can contribute to create ontology from SRS semi-automatically.*
O3. *Develop a framework for creating system specification ontologies from SRS documents.*
  – O3.1. Develop methods to extract terms from SRS, formation of ontology elements such as, concepts and relations, and populate ontologies.
  – O3.2. Develop a proof of concept of the framework to create ontologies from SRS.
O4. *Evaluate the constructed framework and the methods within framework.*

4 Methodology

The following steps will be performed to meet the objectives and reach the aim.

- We will analyze software requirements specifications for a software systems within a particular organization.
- We will develop an ontology manually from the requirements specifications to analyze the ontology development process to make it semi-automatic.
- A systematic literature review (SLR) will be done for ontology learning from SRS to keep updated about relevant research in the field. Another reason for SLR is to investigate existing methods of ontology learning can partially be used to learn ontologies from SRS.
- We will develop a framework and a set of methods within framework and implement to create system specification ontologies (SSOs) from SRS semi-automatically.
- We will do evaluation of the developed framework, such as by using the constructed ontologies in the software testing process and automating test case generation, evaluating through correctness and usability of constructed ontologies.

5 Preliminary Results and Future Work

The current research is part of the OSTAG[2] project and the thesis is in an early stage. The analysis of software requirements specifications and related documents for a communication software is in progress. An ontology has been developed manually from software requirements specifications for the particular communication software. An evaluation of manually developed ontology has been done by using it in the software testing process and by automating test case generation. By developing an ontology manually and evaluating it in the OSTAG project, we have got a clear idea about the type of domain knowledge and information required for an ontology to be useful for test automation. The ontology developed manually will be used for the evaluation of semi-automatically constructed ontologies. Further, a literature review for ontology learning from text has been conducted as software requirements are often written in natural language text.

The next step is to investigate existing methods of ontology learning that partially can be used for creating ontologies from SRS. Afterwards, the development of the framework and a set of methods within framework to perform different ontology-learning tasks, such as terms and concepts extraction, creation of instances and relations between instances, will be developed to learn ontologies from SRS. A proof of concepts will be developed of the framework. Finally, the evaluation of framework will be done through different ways. The evaluation will be done by comparing results of the framework and methods with the existing solutions. An indirect evaluation of the framework will be done by using constructed ontologies in software testing process and automating test case generation.

[2] http://ju.se/en/research/research-groups/computer-science-and-informatics/semantic-technologies/research-projects/ontology-based-software-test-case-generation-ostag.html.

Acknowledgments. I would like to thank and gratefully acknowledge my supervisors, Dr. Vladimir Tarasov, Dr. He Tan from School of Engineering, Jönköping university and Dr. Birgitta Lindström from Skövde university for their support, motivation, valuable comments and guidance. This work is part of OSTAG project, funded from KK-Foundation by grant KKS-20140170 and will be carried at School of Engineering, Jönköping University, Jönköping.

References

1. Hesse, W.: Ontologies in the software engineering process. In: EAI, pp. 3–16 (2005)
2. Happel, H.-J., Seedorf, S.: Applications of ontologies in software engineering. In: Proceedings of Workshop on Semantic Web Enabled Software Engineering (SWESE) on the ISWC, pp. 5–9. Citeseer (2006)
3. Zedlitz, J., Luttenberger, N.: Transforming between UML conceptual models and owl 2 ontologies. In: Terra Cognita 2012 Workshop, vol. 6, p. 15 (2012)
4. Tarasov, V., Tan, H., Ismail, M., Adlemo, A., Johansson, M.: Application of inference rules to a software requirements ontology to generate software test cases. In: Dragoni, M., Poveda-Villalón, M., Jimenez-Ruiz, E. (eds.) OWLED 2016, ORE 2016. LNCS, vol. 10161, pp. 82–94. Springer, Cham (2017). doi:10.1007/978-3-319-54627-8_7
5. Souza, É.F., Falbo, R.A., Vijaykumar, N.L.: Ontologies in software testing: a systematic. In: VI Seminar on Ontology Research in Brazil, p. 71 (2013)
6. Yijian, W., Zhang, S., Zhao, W.: Towards learning domain ontology from legacy documents. In: Fourth International Conference on Digital Society, ICDS 2010, pp. 164–171. IEEE (2010)
7. Zhou, L.: Ontology learning: state of the art and open issues. Inf. Technol. Manage. **8**(3), 241–252 (2007)
8. Wong, W., Liu, W., Bennamoun, M.: Ontology learning from text: a look back and into the future. ACM Comput. Surv. (CSUR) **44**(4), 20 (2012)
9. Fortuna, B., Grobelnik, M., Mladenic, D.: OntoGen: semi-automatic ontology editor. In: Smith, M.J., Salvendy, G. (eds.) Human Interface 2007. LNCS, vol. 4558, pp. 309–318. Springer, Heidelberg (2007). doi:10.1007/978-3-540-73354-6_34
10. Cimiano, P., Völker, J.: Text2Onto. In: Montoyo, A., Muńoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, pp. 227–238. Springer, Heidelberg (2005). doi:10.1007/11428817_21
11. Jiang, X., Tan, A.-H.: Crctol: a semantic-based domain ontology learning system. J. Am. Soc. Inf. Sci. Technol. **61**(1), 150–168 (2010)
12. Drymonas, E., Zervanou, K., Petrakis, E.G.M.: Unsupervised ontology acquisition from plain texts: the *OntoGain* system. In: Hopfe, C.J., Rezgui, Y., Métais, E., Preece, A., Li, H. (eds.) NLDB 2010. LNCS, vol. 6177, pp. 277–287. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13881-2_29
13. Nie, X., Zhou, J.: A domain adaptive ontology learning framework. In: IEEE International Conference on Networking, Sensing and Control, ICNSC 2008, pp. 1726–1729. IEEE (2008)
14. Browarnik, A., Maimon, O.: Departing the ontology layer cake. In: Modern Computational Models of Semantic Discovery in Natural Language, p. 167 (2015)
15. Park, J., Cho, W., Rho, S.: Evaluating ontology extraction tools using a comprehensive evaluation framework. Data Knowl. Eng. **69**(10), 1043–1061 (2010)