CAN ONTOLOGIES BE SUFFICIENT SOLUTION TO REQUIREMENT ENGINEERING PROBLEM?

Richa Sharma[1] and K. K. Biswas[2]
[1]School of Information Technology, IIT, Delhi, India
[2]Department of Computer Science, IIT, Delhi, India

Keywords: Requirement engineering, Requirement engineering problem, Knowledge engineering, Ontology.

Abstract: The growing interest in Ontology has resulted in an increasing interest in Ontology based Requirement Engineering over last few years and a lot has been talked about ontological approach towards the core problem of requirement engineering, i.e. requirements representation and analysis. However, it is important to understand the Requirement Engineering problem and to what extent Ontology can serve as solution to this problem. There are two aspects to the problem – first establishing and evolving Requirement Engineering ontology and second aspect pertains to the Ontology of the business domain in question. In this paper, we discuss the second aspect in detail as requirement engineering is all about capturing, validating and maintaining the requirements for the system in question. We argue that though ontology provides the building blocks for the solution to the requirements problem but the blocks need to be integrated into a process-flow satisfying the quality needs and environmental constraints.

1 INTRODUCTION

The interest in formalizing Requirement Engineering practices has aroused a growing interest in making use of Ontology in the field of Requirement Engineering. Ontology has been listed as a part of metaphysics dealing with questions like what entities exist or can be said to exist, how these can be grouped, related within a hierarchy, and sub-divided according to similarities and differences. Drawing a parallel between philosophy and information science, it can be said that ontology is an explicit specification of conceptualization (Gruber, 1993). Ontology is a representation of knowledge as set of concepts and their relationships; and, can be used to reason about entities, hierarchies and the relationships within that domain. A Requirements Engineer needs to perform similar task while modelling and analyzing the requirements during requirements processing. Ontology offers a suitable solution to the tasks of a requirements engineer. These two fields, namely Ontology and Requirement Engineering (RE hereafter), have a lot in common.

Whilst much work has focused on ontology-based requirement engineering (Dobson and Sawyer, 2006); (Yu, 1997); (Breitman and Leite, 2003) it is interesting to study the role played by ontologies in RE. In this paper, we study if ontologies can be viewed as a solution to the basic RE problem – *requirements representation*. Before arriving at a conclusion, we need to have a careful look at the RE problem and the kind of solution expected. The remainder of this paper is organized as: Section 2 discusses the RE problem. In section 3, we present the relevance of ontology in context of RE. Section 4 presents an illustration with reference to the points studied in section 3. Finally, we conclude on applicability of ontology to RE in section 5.

2 THE RE PROBLEM

The direction of work in RE has varied based on programming practices on one hand to business goals on other hand. These varying directions have resulted in different approaches to RE as: structured RE, object-oriented RE, aspect-oriented RE, goal-oriented RE. Despite their differing nature, the central problem that RE aims at resolving is still the same – the *requirements problem*. Broadly speaking, the *requirements problem* has to deal with *what* is to be represented and secondly, *how* it is to be

Sharma R. and Biswas K..
CAN ONTOLOGIES BE SUFFICIENT SOLUTION TO REQUIREMENT ENGINEERING PROBLEM?.
DOI: 10.5220/0003667204610465
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2011), pages 461-465
ISBN: 978-989-8425-80-5

represented. Let's discuss each of *these two points of requirements problem* in detail below.

2.1 What is to be Represented?

The elicited requirements are a reflection of the real-world system including entities; their properties, and behaviour within the constraints of the domain. Requirements representation needs to cover all the relevant aspects of the real-world under normal as well as exceptional scenarios. There are constraints and the presuppositions that need to find its way in the requirements representation. The analyst usually grapples with the issue of coming up with a requirements representation that portrays the real-world system as close as possible. As suggested in (Zave and Jackson, 1997), requirements problem amounts to finding the representation S, that for given domain assumption K satisfies the given requirements R. If **K, S** and **R** are represented in mathematical logic, then the ***requirements problem*** is solved once the requirements engineer finds S such that **K, S |- R**.

The RE ontology is expected to cover an exhaustive spectrum of real-world system as discussed above. RE ontology itself has been refined and evolved over last few years. In (Jureta, Mylopoulos and Faulkner, 2008) RE core ontology has been revisited to include *goal, softgoal, quality constraint, plan and domain assumption;* and the relationships are *attitude-based* optionality and preference, *justified approximation* and *non-monotonic consequence*. But an object-oriented view of the system is missing here. Since our focus is to examine whether ontology can play an effective role in RE or not, we won't delve deeper into the concern of RE ontology. Nevertheless, this discussion prepares the ground for our arguments in context of ontology's role in RE to be covered in next section.

2.2 How It is to be Represented?

This question is important from the point of view of how the representation is going to be utilized. An answer to this question needs to address the appropriate level of abstraction, generalization and prioritization; and, this appropriate level has to do with the expressivity and reasoning power of the representation. Most of the formalism for requirements representation is often reducible to first-order logic. RML (Greenspan et al., 1986) and Telos (Mylopoulos et al., 1990) representations find their well-formedness and semantics in the roots of first order logic. KAOS (Dardenne et al., 1996) and i* (Yu, 1997) representations are goal-oriented in nature. The advantage of formal representation of requirements lies in reasoning with representations for inferencing purpose. We would like to highlight whatever approach is adopted for requirements representation; the representation should be expressive, flexible and should have sufficiently good reasoning and inferencing power.

3 ONTOLOGY IN RE CONTEXT

In the words of Gruber, ontology is a specification of conceptualization (Gruber, 1993). Ontologies are useful for defining a common vocabulary for a domain, and coming up with a common taxonomy of terms used across multiple stakeholders. We discuss the relevance of ontology in context of RE below:

3.1 Knowledge Capturing

The real-world concepts have been represented using various other methodologies prior to ontological technique like UML class diagrams (Booch et al., 1990), conceptual graphs (Mineau et al., 2000). Ontology also belongs to the family of knowledge representation languages for authoring concepts. With the advent of web ontology language, OWL (OWL), ontologies offer a relatively easy mechanism of sharing information across web. As the nature of RE problem involves capturing and sharing information across multiple interested parties, ontology offers potential use from two points of view - first, ability to define common vocabulary for the represented knowledge and second, the ability to share information in terms of the defined vocabulary. Though ontology serves as a suitable technique for presenting a unified shared view of the world, but is that sufficient to capture the requirements of a real-world system for which an information system is to be developed? We'll explore this question in the next section.

3.2 Limitations

The limitations of ontology in context of RE can be considered in terms of following points:

1. *Dependency on Domain Experts.* Capturing well-defined, ordered and meaningful ontology pertaining to a certain domain requires the involvement of subject-matter or domain experts (SMEs hereafter) to correctly classify and group the entities. Availability of SMEs is a problem and secondly,

ontology-building task in itself is an arduous and time-consuming one. Also, the model cannot be built in one go – it evolves over time with multiple inputs pouring in. Another dependency on SMEs comes from the fact that there are implicit knowledge components in any domain under consideration. These can be in terms of entities classification; association or the rule processing part. Since SMEs are well-versed with the system behaviour, it becomes important to take their views to ensure that modelling is correct and complete.

2. *Exceptions in Hierarchies.* The hierarchies captured through ontologies can possibly have exceptions in terms of attributes or behaviour exhibited by them. Unfortunately there is no way to represent such exceptions or prioritize the ontologies to avoid conflicts. A famous example relevant to this scenario is that of bird ontology where classification into flying and non-flying birds is a problem. Though there are alternatives to capture this behavioural exception but that does not go in line with abstraction properties.

3. *Constraints.* Any business domain under study will definitely be operating under certain constraints related to mode of operation or external factors. Ontologies support expressions for the constraints that are expressible in terms of attributes of entities. But, other constraints like pertaining to the environment or to the mode of operation of the domain are not always expressible as ontology. For example: number of permissible seats to enrol for a course may vary as per the course level, course name, year of admission etc. It is not possible to generalize number of seats at a particular level. Adding this as a constraint would require procedural logic representation to express the constraints. This refers to a situation owing to the mode of operation.

4. *Business Processing Rules.* The business processing rules are an important aspect of the real-world system as rules only govern the behaviour of the entities in a domain; define their tasks and sequence them in a manner that aids in collectively achieving a common goal of the domain under study. Ontology provides the building blocks of the system; but it is the business rules that act as a system integrating agent by binding the system in a way that it doesn't fail and remains operational.

Ontologies are not sufficient to capture business processing rules owing to two main reasons, namely: business processing rules are usually complex in nature; and secondly, they span across multiple entities and different attributes. It is imperative to integrate some means of rule-processing techniques or methodologies with ontology.

5. *Conflicting views.* When it comes to business requirements, users and various other stakeholders can have multiple conflicting views corresponding to a certain processing, or even classification and grouping of entities. There can be conflicting requirements and prioritized requirements. The latter ones refer to trade-off decisions that can be reached to a consensus through ranking or prioritizing the requirements. But conflicting viewpoints for some requirement in a particular release of software need to be addressed while specifying the requirements.

4 AN ILLUSTRATION

We studied the grade-processing part of an educational organization with the idea of expressing the requisite knowledge as ontology only. Though grade-processing part is rule-intensive, we took the challenge of writing the given use-cases using ontological approach only. Our aim in conducting this experiment was to determine if software requirements can be expressed in terms of ontology alone. The experiment started with the identification of entities, relevant attributes and relations in the grade-processing sub-system. We identified a total of 7 entities along with their attributes as:

1) *Student*
2) *Administrator*
3) *Dean*
4) *Course-Coordinator*
5) *Head of Department (HoD)*
6) *Moderating Committee Chairman*
7) *Grades and their sub-categories*

This experiment could not serve as a good example of point 2 discussed in section-3.2 as the only classification was in case of grades with no exception scenario. We'll now discuss our observations for the remaining four points:

1. *Dependency on SME.* Within the periphery of students' grade processing, identifying the ontology was not ambiguous. But, few use-cases presented the instances of implicit/tacit knowledge for which SMEs had to be consulted. Direct Grade Change use-case processing stated: *User initiates 'Direct Grade Change'. User selects academic year, semester and student whose grades need to be changed. User submits the information to save.*

The precondition was: *Grade rules must be defined in the system and grades should be submitted and approved.* And, the post-condition part stated: *Grades submitted are finalized.*

Apparently, there is no problem in this use-case.

But, studying it with other use-cases where three different levels of approval are defined, we needed SME's advice as to understand when can admin intervene for changing the grades? Also, what is meant by grade finalization and if finalization and approval refer to same status? These questions led to the refinement of the values for attribute - *status associated with grades*. The fact that value of an attribute can play a significant role in deciding the operations-flow cannot be unravelled through ontological approach.

2. *Constraints.* The grade master mapping use-case describes as: *User has the option to specify the grade details like grade points and whether it is pass grade or not. The pass grade points vary from A to E and F would indicate Fail. Students' relative scores are mapped to grade points. Course coordinator can alter the mapping under special circumstances and assess the student by asking him to appear for re-exam or viva.*

Expressing the score mappings to grade points with pass or fail constraint was easy to implement as Ontology. But the additional constraints of alteration of mapping by the course coordinator under special circumstances couldn't be captured as representing this knowledge requires procedural logic.

3. Business Processing Rules. The Grade Conversion use-case was defined with processing part stating: User initiates 'Grade Conversion' *process. User selects academic year, semester and course of which he wants to change the grade. User then selects the student names and can optionally give remarks while converting the grade. The updated information is saved.*

The post-condition part stated: *Grade status gets modified and entire workflow of grade approval is initiated.* We couldn't capture this kind of an ordered workflow as represented in the use-case mentioned above using ontology alone.

4. *Conflicting Views.* There were conflicting views on grade conversion process among departments. The rules for grade assignment and grade conversion criterion varied across multiple departments. It is possible for student to be registered in one department and take course in another department – there was conflict in this case which constraint should be applied for grade conversion. We could associate the constraints with each of the departments but could not associate prioritization part between various departments. This particular use-case served an excellent example of implicit knowledge which is there with the people in the system but unknown to the developer.

5 DISCUSSION & CONCLUSIONS

A study of above example shows that ontologies are an expressive means for modelling and designing the semantic structure of the domain under study. But the remaining part of RE ontology - business processing rules, the exceptions and conflicting views is not expressible in terms of ontologies. A lot of focus has been there on ontology based RE but it won't be incorrect to say that the approach is like reinventing the wheel. UML, RML and conceptual graph based entity models are equally good for conceptualising the domain under study. The only advantage of Ontology based approach is in sharing the Ontology across web. It won't be worthwhile to over-engineer if other suitable form of class diagrams already exist or, possibly need not be shared across web. Eventually, requirement analyst would need to integrate the domain ontology model with business processing methodology.

Finally, we would like to conclude with the statement that the approach of ontology-based requirement engineering needs to be substantiated with additional rule-processing mechanism; conflict-handling and prioritization.

REFERENCES

Gruber, T. R., 1993. Toward principles for the design of ontologies usd for knowledge sharing. In *International Journal of Human an dComputer Studies*, *vol. 43*. pp. 907-928.

Dobson, G. and Sawyer, P., 2006. Revisiting Ontology-Based Requirements Engineering in the age of the Semantic Web. In *International Seminar on Dependable Requirements Engineering of Computerised System, Norway*.

Yu, E., 1997, Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In *5th IEEE International Requirements Engineering Conference* , pp. 226-235.

Breitman, K. K. and Leite , J. C. S. do Prado, 2003. Ontology as a Requirements Engineering Product. In *11th IEEE International RE Conference*.

Zave, P. and Jackson, M., 1997. Four dark corners of requirements engineering. *ACM transactions on Software Engineering Methodology*, *6(1)*.

Greenspan, S., Borgida, A. and Mylopoulos, J., 1986. A Requirements Modeling Langugae and Its Logic. In *Information Systems, 11(1)*, pp 9-23.

Mylopoulos, J., Borgida A., Jarke , M. and Koubarakis, M., 1990. Telos: Representing Knowledge About Information Systems. *ACM Transactions on Information Systems*.

Dardenne, A., Lamsweerde, A. van and Fickas, S., 1996.

Goal directed requirements acquistion. *Sci. Comput. Program.,vol 28, no 4*, pp 623-643.

Jureta, I. J., Mylopoulos, J. and Faulkner , S., 2008. Revisiting the core ontology and problm in requirement engineering. In *16th IEEE International Requirements Enginering Conference*, pp . 71-80.

Booch, G., Rumbaugh, J. and Jacobson, I., 1999. The Unified Modelling Language: User Guide*, Addison Wesley*.

Mineau, G. W., Missaoui, R. and Godin, R., 2000. "Conceptual Modeling Using Conceptual Graphs", in *Proc. KRDB*, pp.73-86.

OWL, wikipedia.org/wiki/Web_Ontology_Language