Use of Domain Ontologies to Improve Requirements Quality

Christian Kücherer

Institute of Computer Science, University of Heidelberg
Im Neuenheimer Feld 205, 69120 Heidelberg, Germany
kuecherer@informatik.uni-heidelberg.de

Abstract. [Context and motivation] Requirements are of great importance for the development of software systems to document and meet stakeholder needs. Software requirements can be affected by several quality defects during the Requirements Engineering (RE) process, for example ambiguity, inconsistency, and incompleteness. These quality defects lead to incorrect systems, unnecessary system functions, and thus to additional costs and effort. **[Question/problem]** Domain ontologies (DOs) contain formalized and conceptualized knowledge of real world domains. Existing works show how DOs can be used in RE to improve the quality of requirements wrt. specific quality attributes. During system specification, different specification levels allow explicit decisions about all aspects of the system to be built. It has not been studied so far, how DOs can be used comprehensively on the different levels of system specification and for different quality attributes. **[Principal ideas/results]** The thesis will provide a conceptual framework for utilizing DOs on different levels of system specification. Throughout all DO-based approaches, several implicit DO utilization patterns exist. The framework relates three dimensions: (i) quality attributes, (ii) DO utilization patterns, and (iii) their impact on the specification level. The framework will be evaluated in combination with a task-oriented RE method and a real project. **[Contribution]** This paper describes the problem, related work, main solution ideas, the research methodology, and progress so far.

Keywords: RE quality defects, Requirements Quality Attributes, Ontologies, Domain Ontologies, Task-oriented Requirements Engineering

1 Introduction

Software requirements capture the stakeholder needs wrt. a software system. These requirements are documented in a software requirements specification (SRS) that contains many individual requirements (IRs). During the requirements engineering (RE) activities [9] elicitation and documentation, IRs and SRSs can be affected by quality defects. The ISO/IEC 29148 standard [9] defines that IR must have several characteristics, such as unambiguity, consistency,

completeness, traceability, and verifiability. Characteristics that must be considered for SRSs are, among others, completeness, and consistency. We call these characteristics *quality attributes for IRs and SRSs*. IRs and SRSs cover different levels of system specification, such as supported tasks, user interaction, or the system architecture. Such levels allows to document the decisions from different perspectives onto the system to be developed explicitly.

One way to improve the quality of IRs and SRSs wrt. various quality attributes is the use of *ontologies* [4]. Ontologies conceptualize real world knowledge in the form of machine interpretable concepts, their relations to each other, and their rules [7]. Each concept might be represented by a 3-tuple *subject, predicate, object* describing the relations (predicate) between real world subjects and objects, e.g. (`SRS`, `consistsOf`, `IRs`). An ontology can be *queried* by a formal language such as SPARQL [6] to access its concepts and relations. With *reasoning*, logical inference is supported based on rules inside the ontology. Ontologies are used in Software Engineering (SE) throughout the whole SE process, respectively in all RE-activities for a broad range of problems (cf. Happel and Seedorf [5]). Ontologies can describe SRSs and formal RE knowledge [5,2]. In particular, a domain ontology (DO) describes specific knowledge of a domain. A DO comprise among other things, typical stakeholder roles, functions or tasks, common application components, or entities. An example of a DO is the Semantical Network of Information Management in Hospitals (SNIK) [10], showing concepts and their relations of the information management domain in hospitals[1].

Several works show how RE can be improved with DOs. The utilization of DOs depends on the quality attributes addressed, differs in their prerequisites, and focuses on certain system aspects. However, it has not been studied so far, how DOs can be used comprehensively on different levels of system specification to improve a SRS wrt. multiple quality attributes.

Section 2 of this paper describes the research problem treated in the thesis. Section 3 gives an overview of related work, in particular of existing approaches to utilize ontologies to improve the quality of IRs and SRSs. Section 4 presents the proposed solution. Section 5 discusses the applied research methods. Finally, the paper is concluded with a progress report in Section 6.

2 Problem

A SRS documents various aspects of the system to be developed, such as user tasks/goals, domain data, user interaction, features, data-structures, and software architecture. Existing RE frameworks often group these aspects into *specification levels*. For example, the task-oriented RE framework TORE [1,12] aims to deliver software that satisfies user needs and therefore focuses on stakeholder tasks. TORE encompasses four specification levels: The *goal & task* level focuses on stakeholders, their goals and tasks. The *domain level* accommodates as-is and to-be activities, refining the tasks into subtasks, system features, and domain

[1] A visualization of the ontology is available at http://www.snik.eu/graph

data. The *interaction level* determines how users will be supported in their to-be-activities by the system through use cases, workspaces, user interface-structure, and system functions. Finally, the *system level* determines user interface details and internal system details such as infrastructure and system architecture.

Quality defects can emerge on any specification level. E.g. the task description might miss crucial tasks (incompleteness) or the domain data model might contain redundancies coming from synonyms (ambiguity). DOs can be used during the specification to support the required quality attributes of IRs and the SRS. The DO utilization depends on the required IR/SRS quality attributes, the type of RE-artifacts, and the required level of detail of IRs/SRSs. In consequence, a great variety of DO utilization methods at different specification levels are available. The thesis will provide a framework of the various ways of using DOs on different specification levels to address RE quality attributes. We want to define this framework and evaluate it with the existing RE method TORE.

3 Related Work

There are a several works that deal with RE quality attributes systematically. Wagner et al. proposes the Quamoco Product Quality Modelling and Assessment Approach to close the gap between abstract quality definitions and concrete quality assessment techniques [16]. They relate general quality concepts from existing quality standards, such as ISO/IEC 25010 [8], to specific quality assessment methods. In a mapping study, Pekar et al. [13] investigate the frequency of researched SRS defects and improvement methods in current research. To avoid SRS quality defects, they found several improvement methods, such as correctness- and completeness checking, ambiguity solving, and glossary. Saavedra et al. [15] provide an extensive overview of SRS quality attributes and investigate existing studies to analyze and evaluate quality attributes of IRs and SRSs. They systematically map 15 existing approaches wrt. their impact on 23 quality attributes and show potential influences between quality attributes.

None of the presented works focuses on DO and none investigates the relation to specification levels. We will build on the provided systematics for quality attributes, assessment and improvement methods.

A recent systematic literature review (SLR) by Dermeval et al. [2] investigates the use of ontologies in RE based on 67 studies. The overview shows that ontologies are used manifold in RE, ranging from formalized RE knowledge in ontologies to automatic specification improvement techniques based on reasoning. However, Dermeval et al. do not investigate details on *how* the RE quality defects are addressed by the ontology utilization *exactly*. We therefore are revisiting the 67 studies to go into more detail. In 27 studies we have seen so far that 44 % (12) of the approaches address a single quality attribute only, followed by 25 % (7) which address two. Fifteen percent (4) contribute to three resp. four quality attributes, whilst no approach at all contributes to more than four quality attributes. So far there is no study that relates different ways of DO utilization to comprehensively support various quality attributes.

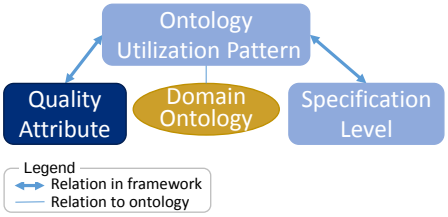

Fig. 1: Framework to relate quality attributes, ontology utilization patterns, and specification level

Qual.Attrib.	**DO Util.Pat.**	**Spec.L.**
completeness	template t.-m. P reasoning P	GT,D
unambiguity	glossary P	GT,D
correctness	formalization P, reasoning P	D

Table 1: Instantiation of the framework: Quality Attribute, DO Utilization Pattern and Specification Level. (GT=Goal & Task Level, D=Domain Level)

4 Solution Idea

The solution idea is to provide a framework that relates *quality attributes*, *DO utilization*, and *specification level*. The benefit of this framework is twofold. First, it supports requirements engineers (ReqEngs) and researchers in the development and customization of RE-methods. Second, it supports ReqEngs in the selection of appropriate methods to match predefined quality attributes in a SRS. This three-dimensional relation is shown schematically in Fig. 1. Thick lines indicate the relations emphasized in the framework, the thin line indicate the relation of the DO utilization to the DO.

In the thesis the details of DO utilization as described in existing studies (selected in Dermevals SLR) will be investigated and synthesized into *DO utilization patterns*. The reason for defining DO utilization patterns is that although existing approaches utilize a DO differently, they share several common characteristics. A DO utilization pattern is similar to a software design pattern [3] and shows a usage scenario of a DO in combination with RE-artifacts, the information flow to a ReqEng, and automatic activities to achieve quality attributes of IRs and SRSs. Such patterns are described graphically. Ellipses indicate ontologies, an actor indicates the ReqEng, a document symbol indicates rules or templates, and boxes indicate automatic activities. Directed arrows between symbols indicate an information flow, non-directional lines indicate an involved-in relation.

There is no catalog of DO utilization patterns so far. Four examples of DO utilization patterns that have been identified so far are shown in Fig. 2. The *glossary pattern* shows a DO that is used by an ReqEng to create a new or to improve an existing RE-artifact, using standard terms of the domain contained in the DO, acting as a glossary. The *template type-mapping pattern* shows that DOs are used in combination with typed templates by the ReqEng to create an RE-artifact. Typed templates can be both, boiler plates that are sentence templates with predefined attributes, or templates such as use-case or subtask-templates. The *formalization pattern* uses a language ontology (containing synonyms), utilize natural language processing (NLP) techniques to create an ontology from existing RE-artifacts and recreate or modify the RE-artifact from this intermediate ontology. Approaches that follow the *reasoning pattern* formalize existing

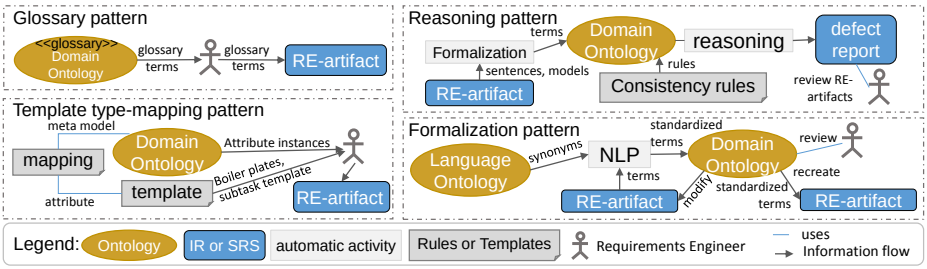

Fig. 2: Identified DO utilization patterns

RE-artifacts into a DO (based on the formalization pattern). With consistency rules and reasoning they identify missing or incorrect requirements collected in a defect report.

Tab. 1 shows the instantiation of the proposed framework with TORE (cf. Sec. 2). To improve completeness, the template pattern can be used on the goal & task level (GT) and the domain level (D) for the complete description of stakeholder tasks and subtasks using terms of the DO. The ambiguity of a SRS can be reduced by an ontology-based glossary, used on GT and D-level, as artifacts on these levels contain domain-specific terms. Correctness of a SRS can be improved by formalizing requirements to the DO and performing automated reasoning of DO with consistency rules. The textual artifacts task description (GT) and subtask description (D), as well as the ER or UML class diagram domain data model (D) are principally suited for formalization.

The following example illustrates the concrete usage of a DO utilization pattern in combination with TORE to reduce ambiguity. Given a DO with the meta model `<role>, <function>, <relation>` and the 3-tuple *(CIO, StrategicPlanning, isResponsibleFor)*, where *CIO* is a `role` and *StrategicPlanning* is a `function`. Further, a subtask (ST) template $\texttt{st}_\texttt{n}$`:(STName, Actor, Description, ...)` allowing any free text, can be mapped to the DO meta model (`STName` → `function`, `Actor` → `role`). For the specification of a concrete subtask, DO concepts as glossary entries can be retrieved by an requirements engineer. The subtask st_1 could be instantiated retrieving the glossary entries *CIO* as `Actor` and *StrategicPlanning* as `STName` that is a function in the DO. Then the fields `STName` and `Actor` of the subtask are filled with glossary entries (*StrategicPlanning*, *CIO*, ...). The *reasoning pattern* can be used to identify other subtasks of the actor improving SRS completeness.

Obviously, DO utilization on various specification levels requires tool support. This tool support may comprise CASE or requirements management tools, an issue tracker, or the ontology editor protégé[2]. The use of these tools in relation to the DO utilization patterns is to be investigated in the thesis.

[2] Protégé ontology editor and framework. http://protege.stanford.edu/

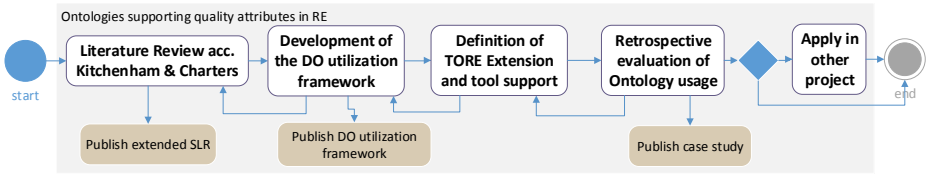

Fig. 3: Research activities

5 Research Methods

The general research method of the thesis is to define the framework based on the extensive **analysis of the existing approaches** that utilize DOs in order to improve IRs and SRSs wrt. their quality attributes, to instantiate the framework for TORE and to evaluate the usefulness of this instance (see also Fig. 3). We plan the application of the framework in a project as far as possible.

First, a systematic **literature review** according to Kitchenham and Charters [11] is performed to extend Dermeval's SLR to answer the **RQ:** *How can domain ontologies be utilized on different system specification levels in an RE method to improve the quality of IRs and SRSs?* To achieve this, all referenced studies of Dermeval et al. are revisited to find *quality attribute*-related patterns in the ontology utilization and understand the principles of existing approaches.

Based on this SLR, the framework is defined by considering related work on quality attributes and improvement or assessment methods. Then, the framework is instantiated for TORE and its different specification levels resulting in different extensions for TORE. They will partly be supported by a tool. As a first step, a JIRA[3] plugin for task and persona description is currently developed in an ongoing B.Sc.-thesis that uses a DO as glossary.

This instantiation will be evaluated in a **case study** using a retrospective evaluation on real RE data of an existing project. In parallel with the previously mentioned SNIK-Ontology, a dashboard-like tool called CIO-Navigator (CION) was developed based on TORE. This software supports the CIO in strategic, tactical, and operational information management decision making. Based on the performed development process and TORE-artifacts and the SNIK-Ontology, the usage of different TORE extensions will be explored. The case study will be organized according to Runeson and Höst [14].

6 Progress

Only the work on the two left activities in Fig. 3 has already started. Approx. 40 % of the SLR-studies have been evaluated wrt. their DO utilization, resulting in four DO utilization pattern identified so far. First ideas for application of the identified patterns for TORE extensions have been developed.

[3] Atlassian issue- and project tracker. https://atlassian.com/software/jira

Acknowledgments I thank my advisor Barbara Paech for her excellent support. This work was supported by the DFG (German Research Foundation) in the Project SNIK, Grant no. 1605/7-1 and 1387/8-1.

References

1. Adam, S., Doerr, J., Eisenbarth, M., Gross, A.: Using Task-oriented Requirements Engineering in Different Domains – Experience of Application in Research and Industry. RE'09 pp. 267–272 (2009)
2. Dermeval, D., Vilela, J., Bittencourt, I.I., Castro, J., Isotani, S., Brito, P., Silva, A.: Applications of ontologies in requirements engineering: a systematic review of the literature. Requirements Eng 21, 405–437 (2015)
3. Gamma, E.: Design patterns. Addison-Wesley professional computing series, Addison-Wesley, Boston, Mass. ; Munich [u.a.], 36. print. edn. (2008)
4. Guarino, N., Oberle, D., Staab, S.: What Is an Ontology? In: Handbook on Ontologies, pp. 1–17. Springer-Verlag Berlin Heidelberg (2009)
5. Happel, H.J., Seedorf, S.: Applications of Ontologies in Software Engineering. In: SWESE'06. pp. 1–14 (2006)
6. Hertel, A., Broekstra, J., Stuckenschmidt, H.: RDF Storage and Retrieval Systems. In: Handbook on Ontologies, pp. 489–508. Springer-Verlag Berlin Heidelberg (2009)
7. Hitzler, P., Parsia, B.: Ontologies and Rules. In: Handbook on Ontologies, pp. 111–132. Springer-Verlag Berlin Heidelberg (2009)
8. ISO; IEC: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models (ISO/IEC 25010:2011(E)) (2011)
9. ISO; IEC; IEEE: Systems and software engineering — Life cycle processes — Requirements engineering (ISO/IEC/IEEE 29148) (2011)
10. Jahn, F., Schaaf, M., Paech, B., Winter, A.: Ein semantisches Netz des Informationsmanagements im Krankenhaus. In: Informatik 2014. vol. LNI P-232, pp. 1491–1498. Gesellschaft für Informatik e.V., Stuttgart, Germany (2014)
11. Kitchenham, B.A., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Tech. rep., Keele Univ.; Univ. of Durham (2007)
12. Paech, B., Kohler, K.: Task-driven requirements in object-oriented development. In: Perspectives on Softw. Requirem., vol. 753, pp. 1–25. Kluwer Academic (2004)
13. Pekar, V., Felderer, M., Breu, R.: Improvement methods for software requirement specifications: A mapping study. QUATIC 2014 pp. 242–245 (2014)
14. Runeson, P., Höst, M.: Guidelines for Conducting and Reporting Case Study Research in Software Engineering. Empirical Softw. Eng. 14(2), 131–164 (dec 2009)
15. Saavedra, R., Ballejos, L., Ale, M.: Quality properties evaluation for software requirements specifications: An exploratory analysis. In: WER 2013. pp. 6–19 (2013)
16. Wagner, S., Lochmann, K., Heinemann, L., Kläs, M., Seidl, A., Goeb, A., Streit, J.: The Quamoco Product Quality Modelling and Assessment Approach. In: ICSE'12. pp. 1133–1142. IEEE Press, Zurich, Switzerland (2012)