A Domain Ontology Building Process for Guiding Requirements Elicitation

Inah Omoronyia[1], Guttorm Sindre[1], Tor Stålhane[1],
Stefan Biffl[2], Thomas Moser[2], and Wikan Sunindyo[2]

[1] Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
{inah.omoronyia,guttorm.sindre,tor.stalhane}@idi.ntnu.no
[2] Institute of Software Technology and Interactive Systems
Vienna University of Technology
Vienna, Austria
{stefan.biffl,thomas.moser,wikan}@tuwien.ac.at

Abstract. [Context and motivation] In Requirements Management, ontologies are used to reconcile gaps in the knowledge and common understanding among stakeholders during requirement elicitation, and therefore significantly improve the quality of the elicited requirements. **[Question/problem]** However, a precondition of state-of-the-art ontology approaches for requirements elicitation is an existing domain ontology. While this is not a trivial precondition, there are only a few reports on approaches to systematically and efficiently build domain ontologies, and these approaches are often highly biased towards their intended use. **[Principal ideas/results]** In this paper, we investigate an approach for building domain ontologies suitable for guiding requirements elicitation. We evaluate the feasibility of the approach based on a real-world industrial use case by analyzing natural language text from technical standards. **[Contribution]** A major outcome is that the proposed approach can help reduce the effort of building domain ontologies from the scratch.

Keywords: Requirements elicitation, domain ontology, semantic analysis, natural language processing, domain engineering.

1 Introduction

The RE process [17] starts out with a specification which is informal, opaque, and dominated by personal views, while the goal is to have a specification, which is formal, complete, and reflects the stakeholders' common view. The use of an ontology can help to tackle these challenges: from opaque to complete because an ontology can encode knowledge about the domain, thus ensuring that important requirements are not forgotten, and from personal to common view because an ontology defines a standard terminology for the domain, which mitigates misunderstandings about terms. If the ontology is defined in a formal language, it will also help regarding the formality

R. Wieringa and A. Persson (Eds.): REFSQ 2010, LNCS 6182, pp. 188–202, 2010.

dimension. There has been an increasing interest in using ontologies to aid the RE process.

Ontologies are specifications of a conceptualization [4] in a certain domain. An ontology seeks to represent basic primitives for modeling a domain of knowledge or discourse. These primitives are typically concepts, attributes, and relations among concept instances. The represented primitives also include information about their meaning and constraints on their logically consistent application [5]. A domain ontology for guiding requirements elicitation depicts the representation of knowledge that spans the interactions between environmental and software concepts. It can be seen as a model of the environment, assumptions, and collaborating agents, within which a specified system is expected to work. From a requirements elicitation viewpoint, domain ontologies are used to guide the analyst on domain concepts that are appropriate for stating system requirements.

There are a number of research approaches to elicit and analyze domain requirements based on existing domain ontologies. For example, Lee and Zhao [13] used a *domain ontology and requirements meta-model* to elicit and define textual requirements. Shibaoka et al. [18] proposed GOORE, an approach to goal-oriented and ontology-driven requirements elicitation. GOORE represents the knowledge of a specific domain as an ontology and uses this ontology for goal-oriented requirements analysis [12]. A shortcoming of these approaches is the need for a pre-existing ontology, as to our knowledge there is no suitable method for building this ontology for requirements elicitation in the first place in an at least semi-automated way. In industrial settings, the task of building domain ontologies from scratch can be daunting, mostly due to the size of technical standard documents that need to be interpreted by domain experts and the wide range of domain concepts coverage that will be the input to such ontologies. Therefore, the domain ontology building task can greatly be leveraged by tool support.

This paper explores the challenge in building a domain ontology that is sufficient for guided requirements elicitation. Firstly, we investigate an approach for building domain ontologies from existing technical standards which the specified requirements need to be compliant with. Our investigation is based on a set of heuristics used for extracting semantic graphs from textual technical standards to generate compatible baseline domain ontologies. Secondly, we present an evaluation of the feasibility of our approach and provide insights on the challenges of semi-automatically building domain ontologies using natural language texts. The remainder of this paper is structured as follows: section 2 presents related work and motivates the research issues; section 3 discusses the characteristics of a suitable ontology for requirements elicitation and also proposes an approach for achieving such ontologies. Section 4 presents the evaluation of our approach and a discussion of lessons learned during this research. Finally, section 5 concludes the paper and presents some ideas for further work.

2 Related Work and Research Issues

Natural Language Processing (NLP) techniques are important when analyzing text to extract domain ontologies for requirements elicitation. NLP generally refers to a

range of theoretically motivated and computational techniques for analyzing and representing naturally occurring texts. The core purpose of NLP techniques is to achieve human-like language processing for a range of tasks or applications [15]. The core NLP models used in this research are part-of-speech (POS) tagging and sentence parsers. POS tagging involves marking up the words in a text as corresponding to a particular part of speech, based on its definition, as well as its context. On the other hand, sentence parsers transform text into a data structure (also called parse tree). Such data structure provides insight into the grammatical structure and implied hierarchy of the input text [1]. Standford parser/tagger[1] and OpenNLP[2] are the core set of NLP tools used in this research. Tag meanings used are from the Penn Treebank project, which involved the annotation of a corpus consisting of over 4.5 million words of English. Words were annotated for part-of-speech (POS) information and skeletal parse structure [16].

Research on domain engineering is also critical to understand an approach to analyze text with the aim of extracting an ontology for requirements elicitation. Domain engineering highlights the process of reusing domain knowledge in the production of new software systems. Domain engineering particularly aims to support systematic reuse, focusing on modeling common knowledge in a problem domain [2]. Sowa's work on conceptual structures [19] introduces a synthesis of logic, linguistics, and Artificial Intelligence as a mechanism for domain knowledge representation.

A closely related research contribution regarding textual extraction of domain ontologies from natural language style requirement documents is the case study on the application of natural language processing to domain modeling presented by Kof [10]. Kof views the domain ontology itself as a valuable and reusable requirements engineering product. He presented three steps for the extraction of a domain ontology which include: *term extraction*, *term clustering and taxonomy building* as well as *finding associations between extracted terms*. The domain model to be extracted is built using the extracted terms as well as the associations between them. Kof [12] also proposed an approach using NLP techniques to construct an initial system model by extracting knowledge from existing requirement texts. Kof [9, 11] furthermore presents mechanisms for analyzing textual scenarios using computational linguistics. The outcome of this analysis process is the identification of whether communicating objects or whole actions are missing in a text. The viewpoint of computational linguistics here is inclusive of NLP and theories about the linguistic knowledge that humans need for generating and understanding written language. Flores [3] has also explored an approach that uses semantic filtering techniques for the analysis of textual requirements descriptions. Flores postulated that filtering relevant text fragments according to semantic criteria enhances large textual requirements description processing. In addition, Flores proposed the use of a linguistic technique known as the *Contextual Exploration Method*, to extract semantically relevant sentences in order to support requirements analysis and validation. Four semantic viewpoints were considered and included: *concepts relationships*, *aspecto-temporal organisation*, *control* and *causality* statements.

[1] http://nlp.stanford.edu/software/lex-parser.shtml
[2] http://opennlp.sourceforge.net/

Although Kof's and Flores' methods are based on analysing natural language texts to extract an a ontology that can subsequently be used for requirement elicitation, no analysis has been done on the suitability or usefullness of the resulting ontology for such purpose. The association of concepts in a domain ontology can be described by its taxonomy or by the use of axioms. The taxonomy is a hierarchical system of concepts, while axioms are rules, principles, or constraints guarding the relations amongst concepts. Furthermore, the level of granularity to which the axioms are specified is highly influenced by its intended use within the ontology [6]. From the viewpoint of using domain ontologies for requirements elicitation, axioms specify the extent to which such an ontology can be useful for the categories of questions to which the ontology can provide answers.

Again, existing related works lack insight into the potential challenges of extracting a domain ontology from a textual source. For instance, such text might not sufficiently describe the domain of concern or contain terms that are not unique to the domain being described. Furthermore, derived ontology from analyzed text can contain unique concepts/relations in the domain of interest which do not contribute to the requirements elicitation process. In such a case, the text analyzed contains valid domain terms that do not necessarily contribute to useful domain ontology. In this research, we reckon that each of these challenges needs to be investigated as to how it can be mitigated.

To address the research issues identified in closely related work, this paper discusses the semantic features of suitable domain ontologies for requirements elicitation and proposes a process for systematic and efficient domain ontology building. We then evaluate the feasibility of our approach based on a real-world industrial use case by analyzing text from technical standards.

3 Ontology Suitable for Guided Requirements Elicitation

For a domain ontology suitable for requirements elicitation, its competence as determined by its axioms is vital. This is particularly true when investigating the use of such ontologies for tracing high-level goals to concrete requirement representations as well as for obtaining insight into the quality of the written requirements. Axioms specified on a relation between two concepts in an ontology aim at providing more meaning to the relation or involving concepts. Richer relational axioms thus suggest more insight on written requirements that reflects corresponding concepts. In this section we highlight three semantic features for enriching the axioms for requirements elicitation ontology.

- *Explicit relational expression*: In addition to the inherent properties of relations such as *transitivity*, *symmetry* and *sub-classes*, requirements elicitation ontologies also aim at associating specific semantic attributes that have domain specific implications. This approach is in contrast for instance to the work of Kitamura et al. [8], who used static predefined stereotypes in naming relations. For example, the relation between the two concepts agent and message using stereotypes is represented by *agent<requires>message*. The predefined stereotype *<requires>* hides the semantic implication of the nature of the relation (e.g., the relation could imply send, receives, blocks etc.). Such otherwise hidden semantics could be useful in

guiding the analyst in determining the relevance of a prescribe trace inference from the ontology to the system being specified.

- *Qualified identification of relations*: Ontologies used to support computers in reasoning will normally identify relations by the use of a single so-called interesting or *performative* verb. These are verbs whose action is accomplished merely by saying them. Performative *verbs* such as *requires, sends,* or *request*, explicitly convey the kind of act being performed by a concept by virtue of an involving relation. But considering an ontology for guided requirements elicitation, the semantic implications of such performative verbs are normally described in an adjoining qualifier such as adjectives and conjunctions. Thus, it is more insightful to name the relation between agent and message using the identifier "periodically sends" rather than only using "send". In this research we explore the use of performative verbs in combination with their qualifiers to semantically identify relations between concepts.
- *Temporal and spatial expressions*: For using domain ontologies for requirements elicitation, we need insight into temporal and spatial implications of relations that exist between concepts. For example, assume A, B and C are concepts in a domain and the description "A requires B during C" is a feature used to characterize a domain. For semantic insight during requirements elicitation, it is important that the explicit relation that exists between A and B as well as the temporal relation that A has with C are captured by the ontology and made obvious to the analyst.

Building domain ontologies with the above semantic features is challenging as it requires domain experts to describe and document their knowledge about the domain with the meaning of concepts and implied relations in a detailed manner, which will be time-consuming. In this research, we explore a rule-based approach that uses NLP techniques to evaluate the possibility of automatically capturing initial or baseline domain ontology from existing text.

3.1 Rule-Based Baseline Ontology Extraction

The basis of this approach is: given some pre-processed textual document and some predefined heuristics based on NLP, it is possible to extract ontology concepts and relations that are semantically meaningful for requirements elicitation. The pre-processing of the document is normally a manual process and ensures that the text from which concepts and relations are to be extracted is suitable for sentence-based analysis. This includes the removal of symbols or formatting from text that will otherwise alter the meaning of extracted concepts or relations. It is worth mentioning that the more detailed a document is pre-processed, the more effective domain ontologies can be extracted from the natural language text. In contrast, for large documents such detailed preprocessing is difficult, as it requires more effort from the domain experts. This challenge for large documents necessitates an (semi-)automated pre-processing approach to help improve the resulting ontology. In the first instance, the rule-based ontology extraction investigates two automated document pre-processing mechanisms known as *bracket trailing* and *bridged-term completion*. Subsequently, Subject-Predicate-Object extraction, association mining and concept clustering is executed on the pre-processed text.

Bracket trailing: Textual descriptions normally use bracket pairs or dashes as punctuation marks to set apart or interject supplementary text within other texts. A common use for brackets in the writing of technical standards is to indicate a reference within the text. They are, however, also frequently used to provide explanatory words or phrases. It is most common for the bracketed text to be used within a single sentence and these texts can be seen as pointers to the concepts represented in a particular sentence [14]. Given the sentence: *"A PLC with a safe transmission protocol (see figure x) shall be restricted to the communication end devices (F-Host, F-CPU, F-Device and F-I/O-Module)"*, the aim of bracket trailing is to filter out pre-determined reference pointers such as "figure x" from existing brackets. Subject/object concepts are then extracted from the remaining text within the bracket. Extracted subjects/objects are finally related to the *head subject* or *object* depending on whether the bracket is used within the noun phrase (NP) or verb phrase (VP) of the sentence. Relations derived via bracket trailing are semantically identified using the stereotype <refers to>. For the example, the concepts "F-Host", "F-CPU", "F-Device" and "F-I/O-Module" are extracted by bracket trailing and related to the concept "devices".

Bridged-term completion: Given the phrase "an input or output device is required by the system", NLP analysis will generate "input", "output device" and "system" as potential concepts. For this example, an understanding of the context of the phrase, suggests that the combined terms "input device" rather than just "input" more completely describe the concept in an unambiguous way. Such ambiguity in concept identification is common in text and it is normally left to the reader of the text to decipher their contextual implications. In this research, an ambiguous term is referred to as bridged-term, indicating that a human interpretation is required to capture its semantic implication. Bridge-term completion involves a semi-automated process of discovering and correcting bridged-terms in textual documents using observed patterns in a sentence parse tree. The occurrence of a learned pattern in an analyzed text raises a flag to the domain expert highlighting possible concept ambiguity and a potential more complete set of terms that better describes the concept.

Figure 1 shows the NP tree patterns in which bridged-terms can occur. The leaf nodes in case 1 consist of a singular noun node or a sequence of singular nouns (NN) nodes separated with commas (,) on the left of *and/or* conjunction. Left of the *and/or* conjunction are a sequence of initial NN nodes where the last node is a NN, plural noun (NNS), proper singular noun (NNP) or proper plural noun (NNPS). On the right hand side of the *and/or* conjunction there are no commas. The NP pattern labeled case 2 is similar to case 1. The core distinction is that in case 2 the first leaf node to the left of *and/or* conjunction can also be a JJ qualifier and there is no comma separating the first two leaf nodes in the sequence. During Bridged-term completion, the NP pattern described in figure 1 is parsed, and a recommended set of more complete terms to describe the subject/object concept in the NP tree is deduced based on node combination (see shaded section of figure 1). The example above corresponds to case 1 and yields "input device" and "output device" as potential subject concepts. A flag indicating possible concept ambiguity with the recommended set of more complete terms describing the concept is then brought to the notification of the domain expert.

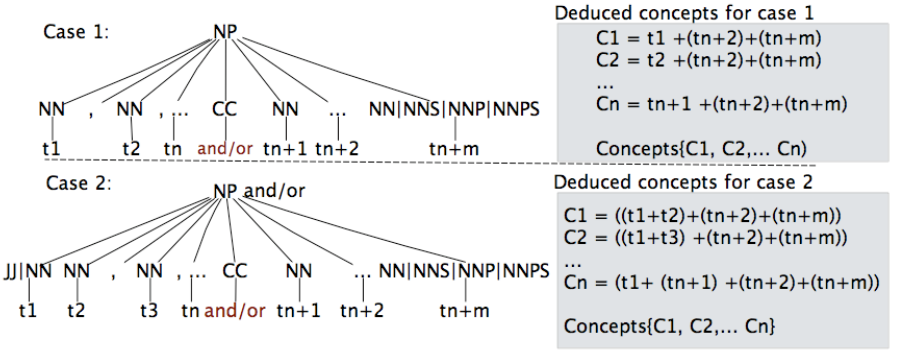

Fig. 1. NP parse tree pattern for bridged-terms

Subject-Predicate-Object (SPO) extraction: SPO is a parsing process that involves navigating the phrase tree of a sentence to extract declarative clauses with their predicate, associated subject and object. Each sentence clause can be said to consist of a noun phrase (NP) and a verb phrase (VP). The NP contains the subject, which can be identified by a noun POS variant (e.g., singular, plural or proper noun) as leaf nodes and can further be semantically qualified by adjectives (JJ) and associated quantities (CD). The VP contains the predicate and the object. The identified subjects and objects are both concepts in the baseline ontology, while the predicate defines the semantics of the relation between the concepts.

When parsing the VP to extract predicate relation, verb variant leaf nodes (e.g., past tense (VBP), present tense (VBZ), etc.) and their first sibling proceeding/preceding qualifiers (e.g. JJ, adverb (RB) and preposition (IN)) are extracted. Similarly, a NP or VP is parsed to extracted noun variation, qualifier and quantity. Extracted nodes are concatenated to a string representation of either the potential predicate relation, subject or object concept. Finally, subject/object concepts extracted during SPO analysis are characterized as head/derived concepts during *association mining*. The *head subjects and objects* act as the domain and range of a relation identified by extracted predicates.

Association mining: Association mining identifies head and derived subjects/objects from the set generated during subject/object extraction. It also extracts other types of relations that are not captured during the predicate extraction process. While the relations between subjects and objects in the predicate extraction are explicitly defined based on the terms that exist in the VP, the relations extracted during association mining are inferred based on any prepositional phrase (PP) whose first preceding sibling node is a noun phrase.

Prepositional phrases normally consist of a preposition and an object of a preposition. Terms that exist in the potential subject/object set and also act as an object in the propositional phrase, are considered as derived subjects/objects. The subjects/objects contained in the first preceding NP are considered as the head subjects/objects. Objects/subjects in a sentence without PP are by default head objects/subjects. From a semantic viewpoint, a preposition is used to illustrate temporal or spatial relationship

between the objects/subjects of the prepositional phrase and objects/subjects of the preceding sibling NP. Such inferred relational semantics are identified and represented by the stereotype <temporally infers> and <spatially infers> respectively. Consider the example above "*A PLC with a safe transmission protocol…*" where the concept "*PLC*" is the head subject while "*safe transmission protocol*" us a derived subject. The preposition "*with*" suggests a spatial relation between the two identified concepts.

Concept clustering: SPO analysis and association mining is likely to result in replication of concepts and relations across sentences. During concept clustering, lexical similarity matching is used to firstly merge the different semantic graphs to eliminate repetitive concepts and relations, and secondly to generate a taxonomy tree based on similarity between terms used to represent different concepts. Vector Space Model techniques have been investigated as the lexical similarity matching approach.

Overall, the rule-based baseline ontology for requirements elicitation involves the following steps: (1) Manual textual document pre-processing to ensure that the text being analyzed is suitable for sentence-based analysis; (2) Automatic bracket trailing filters out predefined reference pointers and extracting relevant concepts referring to a head concept within the sentence; (3) Semi-automated bridged-terms updating to remove ambiguous concepts; (4) Automatic sentence analysis that extracts subjects/objects and predicates as concepts and relation amongst concepts; (5) Automatic association mining to extract temporal and spatial references amongst concepts; and (6) Automatic concept clustering to build a taxonomy of concept.

4 Evaluation and Discussion

This section discusses an initial empirical study of the proposed approach for extracting domain ontologies suitable for requirements elicitation. By using real-world industrial technical standards – in our case from the domains of transport, Adaptive Cruise Control (ACC) - we compare the domain ontology generated by our rule-based approach with a manually generated domain ontology to understand the implications of our approach. We focus on the challenge of irrelevant terms that are not unique to the domain being described and thus do not contribute to the requirements elicitation process. We also evaluate if the different extracted concepts and relations from the analyst, domain expert or via rule-based approach were semantically intuitive and meaningful for guided requirements elicitation. Finally, we present lessons learned.

4.1 Manually Generated and Rule-Based Comparison of Elicitation Ontology

The manual generation of the requirements elicitation ontology involved two experienced participants. The first, who acted as the analyst, had only a vague understanding of ACC but were knowledgeable about requirements elicitation processes, while the second participant had a much deeper insight into the ACC domain and acted as the domain expert. Both participants were knowledgeable of how concepts and relations amongst concepts can be used to generate an ontology.

Paragraphs from two representative sections in ISO 15622 – ACC systems technical standard [7] were presented to both participants (see figure 2 labeled case 1 and 2).

Our selection criterion was a document section that was representative and at the same time provided insights independent of the initial manual pre-processing. This is because the level of manual document pre-processing carried out by a third-party on the selected text can influence the quality of domain ontologies generated using the rule-based approach. Both participants were asked to extract concepts and generate relations among the concepts from the text. To understand the implications of our approach, we feed the same natural language texts into an implemented prototype for automated rule-based ontology approach.

Case 1
The main system function of Adaptive Cruise Control is to control vehicle speed adaptively to a forward vehicle by using information about: (1) ranging to forward vehicles, (2) the motion of the subject (ACC equipped) vehicle and (3) driver commands (see Figure x). Based upon the information acquired, the controller (identified as "ACC control strategy" in figure x) sends commands to actuators for carrying out its longitudinal control strategy and it also sends status information to the driver.

Case 2
Type 1a and 2a ACC systems shall either temporarily suspend operations but remain in the ACC-active or transition to ACC-stand-by if the driver depresses the clutch pedal. For type 2a systems, the Automatic brake maneuver can be continued during the use of the clutch pedal. After the system releases the brakes, the system may either resume ACC control or transition to ACC-stand-by state in response to the driver depressing the clutch.

Fig. 2. Sample text presented to participants (Source: ISO 15622 [7])

Table 1. Number of concepts and relations extracted from sample text

	Concepts		Relations		
	Assumed	Explicit	Assumed	Explicit	Parent-Child
Rule-based	7	30	16	15	21
Analyst	0	26	0	25	0
Domain expert	3	18	8	13	6

Table 1 shows the number of concepts and relations extracted from the sample text. Explicit concepts/relations are directly inferred from the text, while assumed concepts/relations are inferred using reasoned based on concept clustering for the rule-based approach or based on understanding and knowledge of the analyst respectively the domain expert. For each of the categories, a higher number of concepts and relations were identified using the rule-based approach compared to those identified by the domain expert or analyst. Insight from participants showed that since the analyst had a limited understanding of the domain, concept/relation extraction was strictly based on his/her understanding of the sample text. On the other hand, the domain expert relied more on his/her general understanding of the domain to assimilate the meaning and implication of each concept/relation from the sample text. Both participants used one hour to analyze and document a domain ontology based on the sample text. The automated rule-based approach used all the required steps besides the initial manual document pre-processing step. Based on the above sample text, this result is an initial indicator that the automated rule based approach can help reduce the effort in generating requirements elicitation ontology and at the same time achieve a greater coverage of domain concepts. On the other hand, it is also important to understand if

the additional concepts and relations extracted by the rule-based approach are valid, semantically meaningful and necessary and not simply an *over specification*.

Using the ontology extracted by the rule-based approach and manually by the analyst and domain expert, this study focuses on getting insights into the over specification (extracting concepts and relations that are not necessary) or under specification (missing concepts and relations that are otherwise necessary) using our rule-based approach; and if the extracted concepts and relations were semantically intuitive for guided requirements elicitation. We discuss each of these factors and pinpoint how they can be possibly ameliorated.

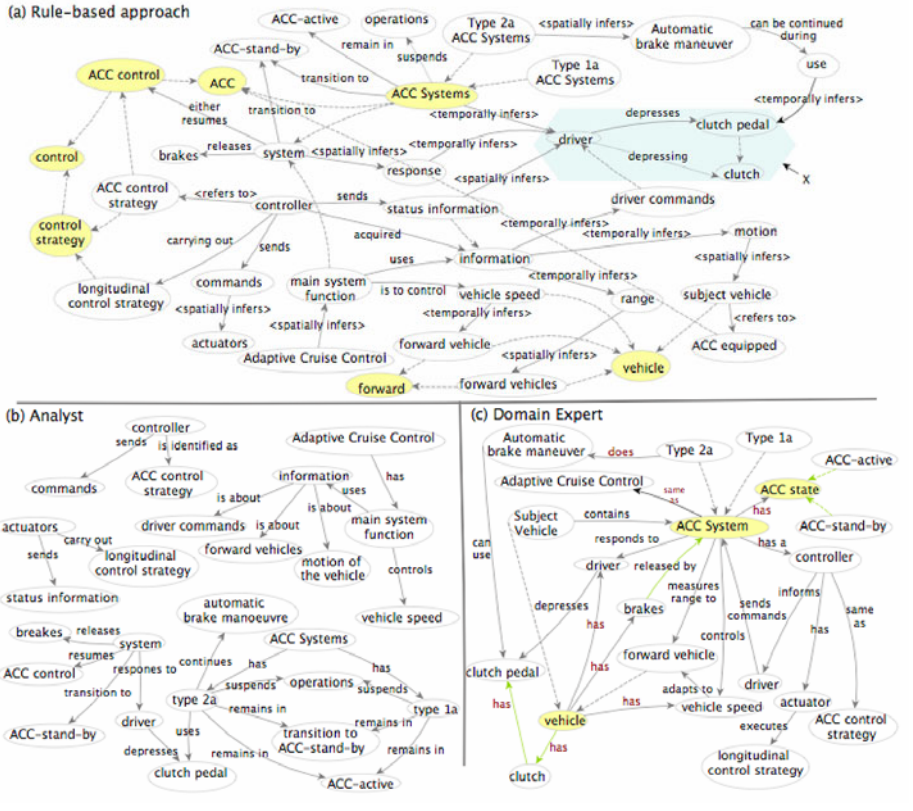

Fig. 3. Rule-based and manually generated ontology

Ontology over specification: It is difficult to state if an ontology is over specified since there is normally no initial understanding of the concepts and relations contained in the domain ontology to be used for a specific requirements elicitation task. Rather, the purpose of this study is to get an understanding of how over specification of concepts and relations can occur for a requirements elicitation domain ontology based on our rule-based semantic analysis of natural language text. Rule-based analysis of the section of sample text "*…adaptively to a forward vehicle by using*

information about: (1) ranging to forward vehicles…" (see figure 2 case 1) results in the over specification of the concepts "*forward vehicle*" and "*forward vehicles*". Apart from the understanding that one concept is singular while the other is plural, the two concepts point to the same semantic meaning. Thus, not much additional insight is obtained by modeling the two as separate concepts. A case of relational over specification using the rule-based approach is demonstrated in the modeling of the relation between "*driver*", "*clutch pedal*" and "*clutch*" concepts (see the shaded section of figure 3a and marked X). In this case, only the relation "*depresses*" between "*driver*" and "*clutch pedal*" is meaningful, while the relation "*depressing*" between "*driver*" and "*clutch*" is not required, given that the former relation implies the later one.

The two cases of over specification pointed out above can be eliminated using stemming/lemmatizing of concept terms to their root words or a more rigorous concept and relational clustering methods. In the first case, the term "*vehicles*" can be stemmed to its root form "*vehicle*", while in the second case, the relation "*depresses*" and "*depressing*" can be merged to a single relation between "*driver*" and "*clutch pedal*" or "*clutch*". On the other hand, modeling of generic terms such as "*control*", "*use*" and "*forward*" in the rule-based approach as concepts can be considered an over specification for domain ontologies suitable for requirements elicitation. Such generic terms are more suitably defined in general ontologies such as Wordnet [3]and can hence be filtered out from specific domain ontologies.

Ontology under specification: Initial insight into under specification when using the rule-based approach can be carried out by checking if all semantically meaningful concepts and relations captured by the analyst and domain expert can be directly or indirectly inferred from the domain ontology generated by the rule-based approach. 24 of the 26 concepts captured by the analyst were also capture by the rule-based approach. As further discussed below, the two remaining concepts "*motion of vehicle*" and "*transition to ACC-stand-by*" is considered wrongfully modeled or less meaningful. 20 of the 21 concepts captured by the domain expert were also captured by the rule-based approach. The remaining concept "*ACC state*" is the concept captured by domain expert which was not represented in the rule-based approach. A follow-up of this finding from the domain expert suggested that "*ACC state*" was informed by his/her understanding that although ACC-stand-by and ACC-active were only mentioned in the text, the two concepts were the possible states of ACC system. Hence the concept "*ACC state*" captured as the parent of ACC-stand-by and ACC-active by the domain expert. On the other hand, it had only been possible for the rule based approach to conceptualize "*ACC state*" if the sample document (case 2 figure 2) was rewritten as "*… but remain in the ACC-active state or transition to ACC-stand-by state…*". In this case, both "*ACC-stand-by state*" and "*ACC-active state*" would be identified and represented by the rule-based approach as concepts while ACC state would be represented as a parent concept.

Three relations present in the ontology created by the domain expert were not captured by the rule-based approach. These include: "*vehicle*" has "*driver*", "*vehicle*" has "*brakes*" and "*ACC system*" same as "*Adaptive Cruise Control*". In all three cases, the sample text was not sufficient and did not contain possible references to suggest

[3] http://wordnet.princeton.edu/

such relation among concepts and was hence impossible to infer for an automated approach. Overall, insights from this study demonstrate that the challenge of under specification using rule-based approach can be reduced by either rigorous manual preprocessing of text document; providing sufficient text for rule-base analysis or by domain experts manually adding the missing concepts and relations.

Semantically intuitive and meaningful ontology: For the ontology generated by the analyst, the concept "motion of vehicle" is wrongly modeled. The reason for this is that in the sample text, the concept "*information*" is precisely related to "*the motion of the subject vehicle*" and not to every "*vehicle*". Similarly, the concept "*transition to ACC-stand-by*" confounds the already modeled relation between "*transition to*" that exist between the concepts "*system*" and "*ACC-stand-by*". The concepts "*actuators*" and "*longitudinal control strategy*" are similarly modeled using the relation "*carry out*" and "*carrying out*" for the domain ontology generated by the analyst and by the rule-based approach respectively. However, from a linguistic viewpoint, the expression "*actuators carry out longitudinal control strategy*" is a complete self-defining phrase, while the expression "*actuators carrying out longitudinal control strategy*" suggests the need for an additional support phrase. In this example, the defined relation between "*actuators*" and "*longitudinal control strategy*" from the analyst is semantically more intuitive than the relation generated by the rule based approach. Such linguistic issues in the definition of relations for the rule-based approach can possibly be reduced if during the predicate extraction phase of the SPO analysis, the root form of the verb gerund or present participle (VBG) is used.

The outcome of the study also suggests that domain ontologies originating from the rule-based approach and from the domain expert should complement each other. This is because concepts and relations are sometime better modeled using the rule-based approach than the domain ontology created by the domain expert, and vice versa. A core observation of the comparison of the domain ontology created by the rule-based approach and the one created by the domain expert is that relations between concepts can sometimes be represented in a rather concise but semantically equivalent and meaningful way. For example, as shown in figure 3a, the relation between the concepts "*Automatic brake maneuver*" and "*clutch pedal*" is identified using an intermediate concept "*use*" with <temporally infers> and "*can be continued during*" relational identifiers between them. In the domain ontology created by the domain expert (figure 3c), a more concise relational identifier "*can use*" links the two concepts "*Automatic brake maneuver*" and "*clutch pedal*". On the other hand, the rule-base approach also demonstrates cases where the use of concepts as intermediaries provides more insights into the relational semantics. For instance, the ontology created by the domain expert (figure 3c) relates the two concepts "*controller*" and "*driver*" via the relation "*informs*". While this is a semantically valid relation, the token that is transmitted from controller to driver is not an explicit characteristic of the relation. Rather, in the rule-based ontology, the "*controller*" and "*driver*" concepts are related via an intermediate concept "*status information*" with <spatially infers> and a "*sends*" relational identifier between them. In this case, the conceptualization of "*status information*" provides more details on the token that is transmitted from the controller to driver.

4.2 Lessons Learned and Limitations of Rule-Based Approach

The core lesson learned in this research is that domain ontologies for supporting requirements elicitation can be achieved by extracting knowledge from technical documents. The domain ontology manually generated by an analyst has shown to be more prone to error when identifying concepts and relations than the ontology that is automatically generated. This is understandable, since analysts usually have no knowledge of the ontology domain. The ontologies created by the rule-based approach and by the domain expert can be used to complement each other. Thus, a viable technique for building requirements elicitation domain ontologies is to generate a baseline ontology using the rule-based approach based on the technical documents and then let it be verified and refined by domain experts.

Furthermore, manual document pre-processing before carrying out sentence based NLP analysis that extracts concepts and relations is critical but in non-trivial cases difficult to achieve. This is because the generated ontology is highly dependent on the quality and format of source text. Our general experience is that domain standard texts tend to conform to good linguistic style and in some cases use controlled language subsets. This is normally not the case when source of the text is informal documents such as emails, interview transcripts and web pages. The successful application of the rule-based ontology generation approach has so far been validated for a domain standard text, and hence might not be a valid approach for informal text sources. Automated document pre-processing such as bracket trailing and bridged-term completion, where possibly ambiguous terms are brought to the notice of the domain expert, are viable options to reducing manual preprocessing effort.

Bridged-terms completion can sometimes raise false alerts. For instance, the sentence "*Safety communication and standard communication shall be independent*" will alert the domain expert on possible concept term ambiguity, even though "*safety communication* and "*standard communication*" are both completely defined concepts. As part of our future work, we plan to investigate a machine learning approach to reduce such false positives. Bracket trailing relies on the assumption that it is common for the supplementary material in a bracketed text to provide more information on the particular single sentence. Such an assumption cannot hold for writing styles where a bracketed text is used to provide supplementary material that references multiple sentences.

As in most text analysis techniques, a 100% precision/recall is difficult to achieve although a high precision/recall rate for the rule-based approach can be inferred for the text used for the initial study. In the first case, a high precision is inferred based on the analysis of sample text for over specification. Two cases of concept over specification were captured out of 37 concepts (95% concept precision). Similarly, two relations were over specified out of 51 relations (96% relational precision). In the second case, a high recall is inferred based on the analysis of sample text for under specification. The analysis showed that 20 of the 21 concepts captured by the domain expert were also captured by the rule-based approach (95% concept recall). Similarly, three relations present in the domain expert ontology were not captured by the rule-based approach (94% relational recall). Given that this is an initial preliminary study using a relatively small subset of technical standard text, more studies will be required to generalize this outcome for a much larger subset.

This preliminary study reveals a scalability concern. Using the rule-based approach, a small snippet of domain standard text can produce large ontology models (figure 3a). An initial insight applying our approach to a larger text suggests that at the early stage, the size of the ontology had a relatively linear growth as text from different sections of the domain standard was analysed. As the volume of text analysed increased, a peak growth is reached when no new concepts were introduced by simply adding text from new sections of the document.

5 Conclusion and Further Work

In Requirements Management, ontologies are used to reconcile gaps in the knowledge and common understanding among stakeholders during requirement elicitation and therefore significantly improve the quality of the elicited requirements. However, a precondition of state-of-the-art ontology approaches for requirements elicitation is an existing domain ontology.

This paper identified three core properties of domain ontologies suitable for requirements elicitation. These include explicit relational expression, qualified relation identification and explicit temporal and spatial expressions. We have investigated a rule-based approach for building such domain ontologies from natural language technical documents. We first introduced bracket trailing and bridged-terms mechanism to help reduce the time and effort that is invested into pre-processing of documents so that they will be suitable for sentence-based analysis. The foundation of this approach lies in the use of NLP techniques to extract subjects and objects as concepts, while the predicate defines the relation between these extracted concepts. Association mining techniques seek to extract other types of relations that are semantically implied in the sentence, but cannot be captured by the predicate extraction process.

We evaluated the feasibility of the rule-based approach based on a real-world industrial use case by analyzing natural language text from technical standards. The study demonstrated that the rule-based approach is a viable technique for building a baseline requirements elicitation domain ontology, which can then be verified and refined by the domain expert. The study showed that requirement analysts were more prone to wrongly identifying concepts and relations to be used in domain ontologies. On the other hand, domain ontologies created by the rule-based approach and the domain expert complement each other. The evaluation provides insights into how over specification and under specification can occur for a requirements elicitation domain ontology based on analysis of natural language text.

In the short term, our further work focuses on getting more insight into the potential of using stemming/lemmatizing to reduce over specification. Future work will also focus on investigating effort reduction measures when extracting requirements elicitation domain ontologies using the rule-based approach. For instance, we seek to understand how domain experts deal with false positive alerts of ambiguous concepts during automated concept terms completion. It is also important to investigate the different modalities for baseline ontology verification and refinement. We are particularly interested in an ontology verification and refinement process that is based on understanding the risk posed by baseline ontology concepts and relations that have not been verified or refined.

References

1. Choi, F.Y.Y.: Advances in domain independent linear text segmentation. In: Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference. Morgan Kaufmann Publishers Inc., Seattle (2000)
2. Falbo, R.d.A., Guizzardi, G., Duarte, K.C.: An ontological approach to domain engineering. In: Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering (2002)
3. Flores, J.J.G.: Semantic Filtering of Textual Requirements Descriptions. In: Natural Language Processing and Information Systems, pp. 474–483 (2004)
4. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquis. 5(2), 199–220 (1993)
5. Gruber, T.R.: Ontology. In: Liu, L., Ozsu, M.T. (eds.) Encyclopedia of Database Systems. Springer, Heidelberg (2008)
6. Ikeda, M., Seta, K., Mizoguchi, R.: Task ontology makes it easier to use authoring tools. In: Proceedings of the 15th International Joint Conference on Artifical Intelligence (1997)
7. ISO standard: Transport information and control systems -Adaptive Cruise Control Systems - Performance requirements and test procedures. 15622 (2002)
8. Kitamura, M., et al.: A Supporting Tool for Requirements Elicitation Using a Domain Ontology. In: Proceedings Software and Data Technologies (2009)
9. Kof, L.: Scenarios: Identifying Missing Objects and Actions by Means of Computational Linguistics. In: Proceedings RE 2007 (2007)
10. Kof, L.: An Application of Natural Language Processing to Domain Modelling - Two Case Studies. International Journal on Computer Systems Science Engineering 20, 37–52 (2005)
11. Kof, L.: Translation of Textual Specifications to Automata by Means of Discourse Context Modeling. In: Glinz, M., Heymans, P. (eds.) REFSQ 2009. LNCS, vol. 5512, pp. 197–211. Springer, Heidelberg (2009)
12. Kof, L.: Using Application Domain Ontology to Construct an Initial System Model. In: IASTED International Conference on Software Engineering (2004)
13. Lee, Y., Zhao, W.: An Ontology-Based Approach for Domain Requirements Elicitation and Analysis. In: Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (2006)
14. Lennard, J.: But I Digress: The Exploitation of Parentheses in English Printed Verse. Clarendon Press, Oxford (1991)
15. Liddy, E.D.: Natural Language Processing. In: Encyclopedia of Library and Information Science, 2nd edn. Marcel Decker, Inc., New York (2001)
16. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: the penn treebank. Comput. Linguist. 19(2), 313–330 (1993)
17. Pohl, K.: The three dimensions of requirements engineering: a framework and its applications. Inf. Syst. 19(3) (1994)
18. Shibaoka, M., Kaiya, H., Saeki, M.: GOORE: Goal-Oriented and Ontology Driven Requirements Elicitation Method. In: Hainaut, J.-L., Rundensteiner, E.A., Kirchberg, M., Bertolotto, M., Brochhausen, M., Chen, Y.-P.P., Cherfi, S.S.-S., Doerr, M., Han, H., Hartmann, S., Parsons, J., Poels, G., Rolland, C., Trujillo, J., Yu, E., Zimányie, E. (eds.) ER Workshops 2007. LNCS, vol. 4802, pp. 225–234. Springer, Heidelberg (2007)
19. Sowa, J.F.: Conceptual structures: information processing in mind and machine. Addison-Wesley Longman Publishing Co., Inc., Amsterdam (1994)