Practical Opportunities in Semi-Automated Requirements Engineering

Kanwal Daud Gill, Arif Raza
MCS National University of Science and Technology (NUST)
Rawalpindi, Pakistan
daud.mscs18@students.mcs.edu.pk, arif_raza@mcs.edu.pk

Abstract: *The critical phrase of Requirements Engineering (RE) has been an active research domain for decades. The evolutions in RE over the years have improved them considerably but still many anomalies exist. However, with emergence and evolution of incredibly exciting fields like Natural Language Processing (NLP), Word Sense Disambiguation (WSD), Artificial Intelligence (AI), Information Extraction (IE) and Text Mining (TM); new possibilities of semi-automated requirements have emerged. The applicability aspect and application opportunities of these domains in RE may differ but certainly their progress has paved way for semi-automation of RE processes. The semi-automation would significantly help in time saving, precision improvement, error and ambiguity handling, and overall process quality improvement. Some of the opportunities, problems and overview of their solutions have been presented in this paper. The meta models of automated requirements recommendation, WSD and to and fro conversion of informal and formal requirements are proposed.*

Keywords: Natural Language Processing, Requirements Automation, Text Mining, Information Extraction, Requirements Process Improvement

Received: 12 March 2014, Revised 3 May 2014, Accepted 8 May 2014

1. Introduction

The area of Requirements Engineering (RE) amongst Software Engineering core phases is a crucial one. Among RE requirements activities, the one named as '*Requirements Elicitation*' is of considerable importance. This is because it is the first activity which consists of stakeholders communicating their wishes to requirements engineer. The effectiveness of elicitation techniques and processes would translate into a development process with much clearer requirements to start with. As a result, the succeeding software development processes would end up with fewer defects and reduced development iterations [1]. Although, gathering requirements may seem a simple ask at first, it has tremendous difficulties and challenges associated with it [2]. The situation becomes even grimmer when internet and computers based requirements elicitation is to be conducted as various advantages of face to face interaction are taken away. The combination of heterogeneous stakeholders, differing mindsets, and distrust pose serious challenges to the overall process. The use of asynchronous and synchronous elicitation and communication means also complexes the process. A number of different elicitation techniques are at disposal of requirements elicitors.

Although a particular requirements elicitation technique and process cannot be called universally useful, as differing scenarios pose the need of choosing differently.

Moreover, the requirements specifications are later on presented either formally or informally. Formally presented requirements

consist of mathematical notations and rule based constructions. Since these formal notions are tough to learn and understand by distantly located developers, especially those working with limited expertise and knowledge [3], the need of informally put requirements arises. These requirements are thus presented through Natural Language (NL), which is easily understandable and is already familiar in usage [4]. The 87.7% of requirements documents are formulated using NL while only 5.3% are written by using formal language [5].

In software development paradigms where internet is involved with segregated stakeholders, the presence of inconsistency, incompleteness and ambiguities strikes. For example, in the case of Open Source Software Development (OSSD), the requirements forums consist of hundreds of thousands of words [6]. These words are all not requirements, but also consist of other informal communication words, e.g. greetings, refutations, negotiations, emoticons, etc. Therefore, even for a good requirements engineer, process of understanding, classifying and clarifying unclear requirements would become a tedious task.

The use of some automated tool and induction of such techniques has the capability of making the life of developers, users and customers easier. There are numerous errors and ambiguities that enter textual elicitation data, but manually identifying and conversely removing them is a tough ask from human developers [6]. The introduction and usage of some automatic techniques would aid the elicitation process. The ambiguities in requirements are the major cause of software project failures, as its resolution is far more expensive [7]. Thus, avoidance and detection of ambiguities is sought as the main problem in requirements elicitation and analysis. It is still an open research problem due to the complexity and diversity of ambiguities encountered in the process.

The rest of this research study is organized as: Section 2 represents the related work. Section 3 glances at the major ambiguities in NL while section 4 closely discusses the ambiguity countering approaches with a critical analysis. Section 5 analyzes the applicability and opportunities of implementing various domains in RE with a narration on attached benefits. Finally, section 6 concludes the study with information regarding future work.

2. Related Work

The innovatively used inter-domain techniques and algorithms can lend good support to the requirements engineering activities. The open discussion forums play a major role in requirements gathering, negotiating, prioritizing and other activities. An interesting approach of identifying requirements from the NL problem statements was proposed by Li et al. [8]. The usage of NLP techniques was made to discover requirements by identification of relations between various grammatical constituents. Cleland-Huang [9] inducted a semi-automatic method for purpose of identifying and classifying the requirements from unstructured as well as structured documents. In another research work Cleland-Huang et al. [10] made use of data clustering techniques in the forums for optimize threads into highly focused ones. Later on a researched framework by Castro-Herrera et al. [11] had data mining techniques in it to collaboratively cluster and filter requirements, stakeholder profile creation and recommender model to support the entire process. The produced prioritized requirements had their rationales attached to aid the stakeholders in achieving shared understanding. Afterwards, Vlas and Robinson [12] developed an automated requirements classifier that helped developers in gaining perspective on what classes of requirements are common in software development. Beg et al. [13] designed a method of dealing with the problem of requirements ambiguities through use of parts of speech tagger. Sentence structures were defined for ambiguity avoidance. Lastly, Sateli et al. [14] established an introductory tool by inducting basic NLP services like semantically enriched meta data support, readability assessment for users, information extractors and writing quality assessment. The usability aspect of this exercise was also tested for judging its wider user adaptability.

3. Natural Language Ambiguities

The term '*ambiguity*' can be defined differently due to its wide impact and conceptual broadness. But it is undoubtedly considered an important aspect of computational linguistics. Ambiguity is termed as a scenario where one statement provides two or more interpretations [15]. Its detection, avoidance, elimination is therefore targeted by research community from decades [16]. The classic methods of requirements clarifications embed some mechanism of involving the customer in the process. The reason being, that only the customer can have final say on what he wants the system to do. Alternatively, any form of intelligent referral system must include contextual knowledge for identifying discrepancies and recommending changes [17].

A number of ambiguities may occur in the NL requirements as 42 types of ambiguities are identified by researchers [18]. However, the NL requirements mainly suffer from three types of ambiguities namely; Lexical Ambiguities, Syntactic Ambiguities

and Semantic Ambiguities.

Lexical ambiguities cause problems due to use of misplaced words in the sentences or phrases that make up the NL requirements. Syntactic ambiguities arise when a sentence may be parsed in more than a single way. Semantic ambiguities surface when the contextual interpretations of a sentence might produce different meanings for different people [15].

The asynchronous communication methods in case of internet based requirements development may take so much more time if frequent clarifications are to be made.

4. Ambiguity Countering Approaches

As there are many forms and kinds of ambiguities that exist in NL textual requirements. So, many approaches for helping in ambiguity reduction have been researched over the years. These approaches can be grouped into four categories, described as:

1. Stakeholders can learn to write in a less ambiguous manner by not using the sentences and grammar structures that can lead to ambiguities [19], [20].

2. Stakeholders can be manually guided and trained to detect the ambiguous requirements constructions [21], [22]. Tool support can also be provided in this regard [23], [24].

3. Restrictions can be imposed on the usage of certain sentence structures that can produce ambiguous requirements [25].

4. Another adopted approach is to restrict the words and phrases that are considered as subjective in a domain [13].

All of these approaches have their shortcomings and advantages. The first approach requires human users to learn to write less ambiguously but it would definitely require a lot more time and training on part of stakeholders. Moreover, there is no guarantee that their writing and communication skills would improve to a particular set mark. This stops the requirements analysts and engineers from adopting this very approach. Moreover, learning the grammar structures is particularly difficult for the countries where English is spoken as a second language. As in manual guidance of second approach also, the human intelligence and judgment gets involved. It thus does not ensure that all ambiguities would be detected as the perception, experience, technical expertise, etc. All are significant contributing factors in this regard. However, tool support for ambiguity detection might produce better results as a tool would work according to its predefined heuristics. The third approach involves the use of Controlled Natural Language (CNL). CNL can restrict the user from writing the requirements or sentence structures that can possibly lead to ambiguities. For example, it may involve putting restrictions on the usage of passive voice, restricting the number and structure of nouns appearing in a sentence, etc. Such usage of CNL would produce better results but making requirements writers or developers use it would be a challenging task. It is somewhat useful in confined software houses where certain standards can be applied and strictly followed but binding the freelance community, Global Scale Development (GSD) developers and Distributed Software Development (DSD) developers etc. would be a tough and costly affair. The fourth approach would produce good results if a study of ambiguities occurring in a specific domain is firstly made. Afterwards, the problematic words, sentences, etc. can be filtered so that subjectivity and ambiguity factors remain low. But this would not produce a tool or method that is universal in applicability. Firstly, identifying the problematic terms would be required each time for all development projects. Therefore, the usage of such a tool or method cannot be made universal for all projects from any domain.

5. Discussion – Involving Automation in Re Problems

The use of fully automated system cannot be called reliable in any way as requirements engineering is a human centric process. The usage of four approaches described above does not involve automated processing, aiding and disambiguation on a large scale. Although they may include some aspects or minimal data mining algorithms but other relevant domains still need to be explored in this regard. The effective use of computer and internet based requirements engineering has been made for purpose of automated discovery, classification, recommendations, stakeholder profile creations, etc. [9], [10], [11], [12], [13]. However, there is quiet less work done in the area of automated requirements ambiguity prevention, detection, and removal. One reason of lesser attempts of making the ambiguity elimination process automatic has its roots in the current acceptability fact that it is impossible for machines to fully understand the NL requirements [26]. The NL requirements embed specific domain knowledge, terminologies and perspectives attached to them.

The basic challenge that lies ahead and requires much needed research and development is of semi-automatic tool support for

requirements engineering. Although the process of requirements elicitation and analysis in particular are human centric processes, but automated aid and support in these phases can significantly leverage the computer based elicitation and analysis. The present elicitation techniques can be redesigned and improved where recommendations, integrated analysis and data based decision making support can be provided. An exciting possibility might be to accept the stakeholder profile information, domain, contextual knowledge, business goals, etc. and recommend a set of elicitation techniques along with other process information. It may include the NLP algorithms supplemented by Text Mining techniques and AI based intelligent agents.

Figure 1 overviews a meta model of a semi-automatic elicitation technique recommender system that may intake stakeholder profile, business goals, domain knowledge and contextual information, etc. After application of NLP, AI, TM, etc. techniques the final output may be a set of recommended elicitation techniques to be adopted for a particular software development project.

Moreover, the activity of requirements verification can be supplemented with excellent and reliable automatic mechanism. Such a mechanism can identify the potential sectors of requirements or NL text which should be quantified and testable but are supplying some metric of requirement completeness. For instance, some intelligent mechanism to detect the emergent properties can be planned and defined. The emergent properties are mainly dependent on the system architecture that consists of contextually appropriate components. The usage of AI mechanisms can intelligently predict the emergent properties that can surface once the system architecture is in place. Moreover, a crucial component of requirements elicitation is '*requirements sources*'. The non-human requirements sources may include goals, domain knowledge, organizational structure and environment along with other factors [27]. The presence of a semi-automated tool may look to highlight the potential requirements from goal statements, domain information, etc. The important entities and heavily contributing factors may be recommended to help the requirements engineer who wishes to get some help in this regard.

One of the major problems with requirements elicitation is the abundance of textual NL ambiguities that enter the system. Some basic suggestions for the use of NLP in RE have been outlined [28]. However the NLP algorithms have not yet become a mandatory part of computer based requirements management systems. The main reason for this is the relative immaturity and some degree of uncertainty associated with this field. However the mechanisms, relevant interpretations, textual transformations, augmentation, etc. can be explored for RE activities. For instance, the NLP algorithms for conversion of formal to informal language requirements and vice versa can be developed. This would significantly help the software community, as on maturity the user would be able to write requirements in NL format that can be translated to formal language requirements. These formal language requirements can in turn be used for further analysis and interpretations as they are easier to understand by machines.

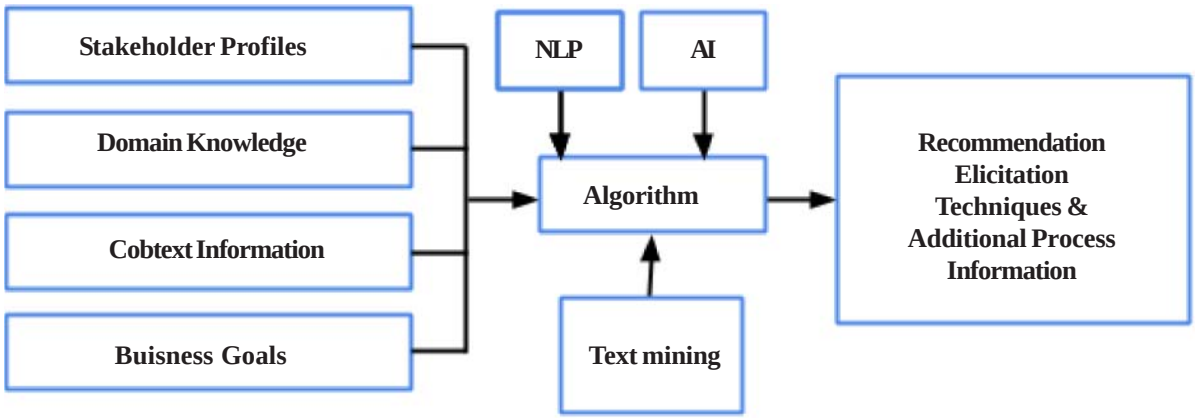

Figure 1. Meta Model of Semi-Automatic Elicitation Technique Recommendation System

Figure 2. Meta Model of to and fro conversions of Formal and Natural Language Requirements

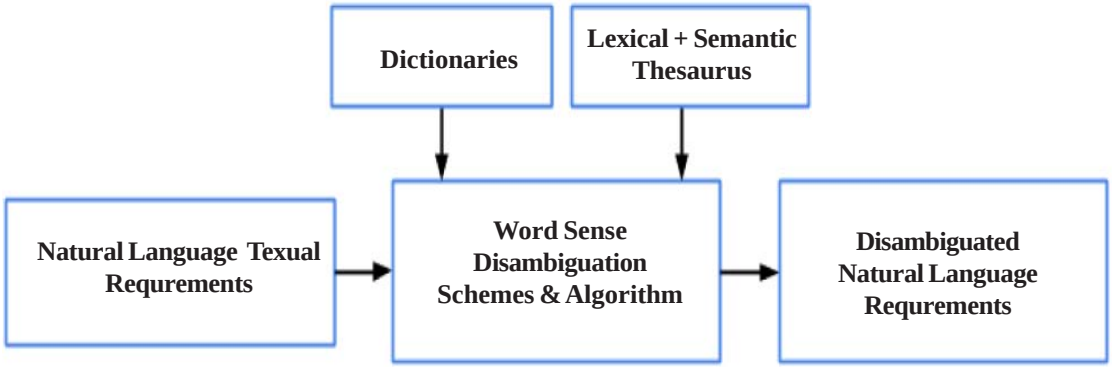

Figure 2 presents a meta model in which specific algorithms using WSD, NLP, TM and AI can be used to convert between notations centric formal and NL requirements.

Moreover, the WSD can singlehandedly be reviewed too for disambiguation of words basing on their specific context.

The usage of algorithms and techniques specific to NL textual problem solving can be developed and embedded in order to improve the requirements. The words with differing senses can be removed or replaced with more accurate words making requirements.

Figure 3 depicts a meta model of WSD system that can be built by applying lexical and semantic resources such as WordNet, FrameNet, etc. for the purpose of producing disambiguated requirements out of anomalies constraining NL textual requirements.

However, WSD algorithms to date are not considered as totally reliable however supervised learning approaches are considered as the most successful ones. But it can still aid the RE processes in sense making and shared knowledge. Moreover, the requirements discovery can be made much better by the usage of existing IE algorithms and developing new ones that are more targeted at requirements engineering. For this purpose, important meta data about requirements and various trends, attributes, etc. can be identified. The linkages between requirements and entities can be established also through some mechanism. This in turn would allow the possibility of suggesting an elicitation technique and criteria for further investigation if necessary. Moreover, automated mind map creation can be supplemented too if entities and classes are discovered relatively accurately. Moreover the thesaurus, terminology generation and identification, key phrase extractions can also be significantly improved. The structured representations (e.g. for databases) can be produced from unstructured textual documents.

TM's existing algorithms and techniques can be curbed to meet the needs of RE where the expected and unexpected connections can be identified from humans' provided textual knowledge. Efficient classification of requirements is one area that can Semantics have proven to be an important aspect in web and other areas. However, by supplementing RE with semantically rich meta data and semantic algorithms, desired results on redundancy countering in requirements can be produced. The problem of redundancy is a problem especially in online requirements systems. Automated domain ontology generation from requirements can be built. Presently, ontology can be generated only for a specific domain which is problematic since it does not allow the opportunity of its universal usage across many domains.

Interestingly the role of statistics for its innovative usage in algorithms cannot be undermined. These techniques can also significantly help and aid the semi-automated requirements elicitation, analysis and broadly the management systems.

6. Conclusion

The RE processes and activities can be improved many fold by induction and development of techniques and algorithms especially designed for this purpose. The problem of wasteful information can be eliminated from NL requirements along with better procedures for requirements analysis as well. The RE process models can be improved and as a result quicker generation of requirements would be made possible. The semi-automated requirements however need to be as accurate as possible. The NLP based precision and recall rates must be high so that maximum of benefit and trust can be put into the computer based semi-automated requirements development and management systems. The quality of newly built systems can be verified for their

effectiveness on requirements verification, validation and software build quality.

7. Future Work

We are currently working on a framework based tool for eliciting requirements in a more comprehensive and OSSD contextual ambiguities avoiding manner. Highly optimized set of suggestions provision to OSSD community is the targeted goal in tool. The transition of informally put NL requirements through an especially designed process into some form of formalized set of requirements is also targeted.

References

[1] Extreme CHAOS, The Chaos Group, (2001). [Online]. www.standishgroup.com

[2] Davis, A., Dieste, O., Hickey, A., Jurist, N., Moreno, A. M. (2006). Effectiveness of requirements elicitation techniques: Empirical results derived from a systematic review, *In*: Proceedings of the 14th IEEE International Requirements Engineering Conference.

[3] Levson, N. (2000). Formal specification: a roadmap, in International Conference on Software Engineering Proceedings of the Conference on The Future of Software Engineering.

[4] Bollinger, T., Nelson, R., Self, K. M., Bollinger, T. (1999). Open-source methods: Peering through the clutter, in IEEE Software.

[5] Mich, L., Franch, M., Inverardi, P. L. N. (2003). Requirements Analysis using Linguistic Tools: Results of an On-line Survey," in 11th IEEE International Conference and Workshop on the Engineering of Computer-based Systems.

[6] Laurent, P., Cleland-Huang, J. (2009). Lessons learned from open source projects for facilitating online requirements processes, in Requirements Engineering: Foundation for Software Quality.

[7] Wiegers, K. E. (2003). Software Requirements, Microsoft Press.

[8] Li, K., Dewar, R. G., Pooley, R. J. (2004). Requirements Capture in Natural Language Problem Statements, Citeseer Technical Report.

[9] Cleland-Huang, J. (2006). The Detection and Classification of Non-Functional Requirements with Application to Early Aspects, in Requirements Engineering, 14th IEEE International Conference.

[10] Cleland-Huang, J., Dumitru, H., Duan, C., Castro-Herrera, C. (2009). Automated Support for Managing Feature Requests in Open Forums, *In*: Communications of ACM.

[11] Castro-Herrera, C., Duan, C., Cleland-Huang, J., Mobasher, B. (2009). A Recommender System for Requirements Elicitation in Large-Scale Software Projects, *In*: Proceedings of the ACM symposium on Applied Computing.

[12] Vlas, R., Robinson, W. N. (2011). A Rule-Based Natural Language Technique for Requirements Discovery and Classification in Open-Source Software Development Projects, in System Sciences (HICSS), 44th Hawaii International Conference.

[13] Beg, M. R., Abbas, D. Q., Joshi, A. (2008). A Method to Deal with the Type of Lexical Ambiguity in a Software Requirement Specification Document, in First International Conference on Emerging Trends in Engineering and Technology-IEEE Computer Society.

[14] Sateli, B., Angius, E. , Rajivelu, S. S., Witte, R. (2012). Can Text Mining Assistants help to improve Requirements Specifications, *In*: Proceedings of Mining Unstructured Data.

[15] Ceccato, M., Kiyavitskaya, N., Zeni, N., Mich, L., Berry, D. M. (2004). Ambiguity Identification and Measurement in Natural Language Texts, Technical Report.

[16] Brill, E., Mooney, R. J. (1997). An Overview of Empirical Natural Language Processing, 18, *AI Magazine*, p. 13-24.

[17] Tjong, S. F. (2008). Avoiding Ambiguity in Requirements Specifications, PhD Thesis.

[18] Berry, D. M., Kamsties, E., Krieger, M. M. (2003). From contract drafting to software specification:Linguistic sources of ambiguity, Technical Report, University of Waterloo.

[19] Dupre, L. (1998). Bugs in Writing: A Guide to Debugging Your Prose, second edition ed., MA, USA: Addison-Wesley.

[20] Berry, D. M. Kamsites, E. (2005). The syntactically dangerous all and plural in specifications, *In*: IEEE Software.

[21] Kamsites, E., Berry, D. M., Paech, B. (2001). Detecting ambiguities in requirements documents using inspections, *In*: Proceedings of the First Workshop on Inspection in Software Engineering.

[22] Kamsites, E. (2001). Surfacing Ambiguity in Natural Language Requirements, PhD Thesis.

[23] Osborne, M., MacNish, C. (1996). Processing natural language software requirement specifications, *In*: Proceedings of the International Conference on Requirements Engineering.

[24] Mich, L., Garigliano, R. (2000). Ambiguity measures in requirement engineering, *In*: Proceedings of International Conference on Software-Theory and Practice, Beijing.

[25] Fuchs, N. E., Schwitter, R. (1996). Attempto controlled english, in 1st International Workshop on Controlled Language Applications.

[26] Ryan, K. (1993). The role of natural language in requirements engineering, *In*: Proceedings of IEEE International Symposium.

[27] Guide to Software Engineering Body of Knowledge, IEEE Computer Society.

[28] Huyck, C. R., Abbas, F. (2000). Natural language processing and requiremnents engineering: a linguistics prespective, *In*: Proc 1st Asia-Pacific Conference on Software Quality.