Automated Requirements Engineering Framework for Agile Development

Muhammad Aminu Umar
Department of Informatics, King's College London
Strand, London WC2R 2LS, United Kingdom
e-mail: aminu.umar@kcl.ac.uk

***Abstract*—Requirements engineering has been established as a critical success factor for software projects. On the other hand, most requirements documents are often written in natural language; often prone to structure errors and inconsistent semantic, thereby, exposing the documents to misunderstanding arising from undue misinterpretations. This paper first proposes a framework to automate requirements engineering activities with focus on modelling while equally articulating the strategies and work plan for the implementation and evaluation.**

Keywords-Automation; Requirements engineering; NLP; Agile development.

I. INTRODUCTION

In software development, Requirements Engineering (RE) is a critical success factor as well as a complex process [1]. It is critical because the quality of the system depends largely on the quality of the requirements and, it is complex because it considers diverse product demands from a diverse set of stakeholders [2]. Inadequacies resulting from the RE process can have negative impact on the overall software development and lead to high costs for any organization involved [3]. Therefore, Requirements Engineering is highly significant to modern success of quality software development.

Software requirements are often specified in Natural Language (NL). Meanwhile, software requirements specified in natural language often suffer from ambiguity, incompleteness and inconsistency in the choice of syntax. Moreover, anything described in NL has the potential and tendency of being influenced by geographical, psychological and sociological factors in terms of understanding and interpretation [4]. It is therefore imperative that empirical studies are conducted to derive solutions to this challenge in software development with the emerging trends in human activities which must eventually be captured beyond NL. Consequently, addressing these problems through researches has paved the way for model-driven engineering and lately automated requirements engineering which have all improve the status of requirements written in natural language. These tools and technologies now facilitate software development generally. More specifically, these tools and technologies enable numerous RE activities such as requirements classification [5][6] requirements validation and review [7][8], inconsistency check in requirements [9], duplicate requirements detection [10], automated RE reuse [11], recommendation of omitted steps in requirements analysis [12] and RE security enhancement [13].

As part of the effort to support agile RE and the development of fast software in general, this work seeks to propose an automated RE framework that combines the Natural Language Processing (NLP) and the Artificial Intelligence techniques for more efficient agile software development. In agile development, rapid change and flexibility are very important and the development process is made as lightweight as possible. The essence of automated RE is to reduce software development time, effort, and cost of RE whilst still maintaining consistent, accurate and comprehensive requirements.

The remainder of this paper is structured as follows: Section 2 discusses automated requirements engineering. Section 3 discusses previous literature on automated requirements engineering. In Section 4, the techniques for the integration of NLP and Artificial Intelligence (AI) are highlighted. Section 5 summarizes the proposed work plan. Section 6 concludes the paper.

II. AUTOMATED REQUIREMENTS ENGINEERING

Requirements engineering is the lifecycle stage with the highest influence on the quality of a final product [14]. Traditional RE process continued to be applied to manage the knowledge generated in this field, and this has made it difficult to attain a quick and objective understanding of the diverse and continuous emerging needs of the interested stakeholders. Apart from the fact that automated RE development reduces cost, effort and time it enhances the quality of the final product. Requirements identification and requirements classification are two important activities supported by automated requirements elicitation from NL document [15]. Some of the successful techniques that have been employed to implement automated requirements elicitation are NLP and Information Retrieval [16]. Additionally, in recent times, text mining, which is an automated technique for generating requirements document have been employed [17].

Requirements engineering involves the process of finding out, analyzing, documenting and checking the services and constraints on a system [18]. The following are generic activities (phases) common to all RE process.

- Requirements elicitation. The process of identifying and collecting requirements from stakeholders and other sources. This includes both functional and non-functional requirements.
- Requirements analysis and specification. It is the logical breakdown and structuring of the proceeds from elicitation. It includes detailed understanding of the requirements and structuring such

 ISBN: 978-1-61208-827-3

information and other derived requirements as written documents and model diagrams.

- Requirements validation. Requires that the collected information is correct and well arranged to meet the system business objectives. This is done by making sure the documents and/or models with specified requirements are accurate, complete and correct.
- Requirements management. This step helps to keep track of possible changes in requirements and ensures that the changes are made to meet stakeholder's requirement.

Automated RE support has been successfully achieved in various activities (phases) of requirements engineering (see Table 1). The NLP plays a great role in achieving automated RE due to the fact that most requirements are written in natural language.

TABLE I. APPLICATION OF NLP IN RE ACTIVITIES

Ref	Elicitation	Analysis	Validation	Management
[12]			*	
[17]		*		
[19]			*	
[4]		*		
[20]	*			
[21]		*		
[22]		*		
[11]				*
[10]				*
[23]				*
[24]		*		
[25]	*			
[26]		*		
[27]	*			
[28]				*
[29]	*			
[30]		*		
[31]				*
[32]		*		

Accordingly, Lucassen et al. [30] have categorized the fundamental approach of all NLP and RE tools into four types based on their functions or what they can do:

1. Finding defects and deviations in natural language (NL) requirements document;
2. Generating models from NL requirements descriptions;
3. Inferring trace links between NL requirements descriptions and
4. Identifying the key abstractions from NL documents.

From the foregoing, the second category of tools forms the concern of this work.

III. RELATED WORK

Automated requirements engineering has attracted the attention of researchers over the years. Each work cited here has proposed an approach or a software tool to achieve the continuous effectiveness and efficient application of automated requirements engineering.

For instance, the UML model Generator from Analysis of Requirements (UMGAR) [4] is a domain independent tool which generates use case diagram, conceptual model, collaboration diagram, and designs class model. The tool follows the object-oriented analysis design approach while eliciting object from requirements described in Natural Language. It also uses the NLP techniques to process textual documents and XML import facility to visualize UML diagram.

Other works, such as [26], have proposed a framework that provides the requirements engineers with the capacity to automatically generate UML class diagram. The framework allows for reinterpretation of natural language requirements into models and has the possibility of specification reusability through reverse engineering process. The approach used the MIMB tool to transform the XML schema into a UML class diagram and vice versa.

In [33], the authors proposed a requirements model generation to support requirements elicitation from a lightweight textual document. However, the approach transforms requirements specification expressed in natural language into semi-structured specifications. The proposal was based on Cerno, which is a semantic annotation environment that uses high-speed context-free robust parsing combined with simple word search.

For [21], the Natural Language Processing technique was applied to automatically transform user stories into UML use case diagram. The approach uses TreeTagger parser to generate use case and Part of Speech (POS) tags allows for the categorization of term into various part of speech.

In [30], a Visual Narrator tool is described as a tool which automatically generates a conceptual model from a collection of user story requirements. This narrator tool was part of what the authors termed the 'Grimm method' – the method combines three Natural Language Processing and thereby enabling requirements tools to support conducting user story-based requirements engineering.

A system described in [22] is claimed to facilitate automatically the creation of conceptual model from functional requirements written in natural language. The tool allows for automatic identification of classes and relationships and subsequently renders the conceptual model with the Extended Entity Relationship (EER) diagram notations.

The use of Artificial Neural Network (ANN) for extracting actions and actors from requirement document by

[32] proposes an approach to automatically identify actors and actions in natural language-based requirements with the goal to overcome the challenges of manual extraction. This tool uses an NLP parser with a general architecture for text engineering, producing lexicons, syntaxes, and semantic analysis. This was achieved through the development of an ANN using five different use cases.

From the reviewed works, it is evident that automation of requirements engineering is a challenging area due to the nature of the diverse inputs to be processed i.e. natural languages. Therefore, this work proposes an alternative automation paradigm framework to automate RE. The proposed approach is an interactive tool support with dynamic model generation. This will be achieved through the integration of NLP and other artificial intelligence techniques.

IV. PROPOSED APPROACH

From our findings of the literature review, there is the need for research into a full and alternative automation paradigm and integration of more artificial intelligence techniques into automating requirements engineering. Therefore, this work proposes a framework that will facilitate automation of requirements engineering activities with specific focus on model generation (modelling) with traceability. The work intends to employ natural language processing techniques and artificial intelligence in the design and development of the framework.

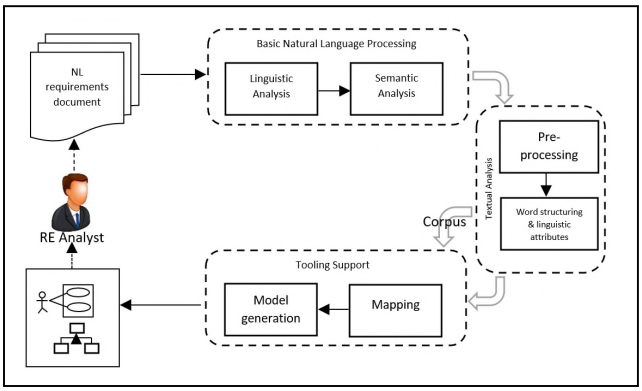

Figure 1. Overview of the proposed framework.

The proposed approach works as follows (see Figure 1). Initially, the analyst (requirements engineer) input a set of textual requirements written in natural language. These texts are analyzed by several natural language processing components with the target of gathering knowledge about the textual scenarios. Sentences, token boundaries, token properties, and semantic constituents are some of the information of the natural language processing components.

After the basic natural language processing is completed, the approach will carry out textual analysis through pre-processing, word structuring and linguistic attributes. The aim of pre-processing is to code the inputs required by the next processing step. A corpus which contains English words with their respective code will be used to compose requirements documents. These are words collected from several English natural language requirements texts from different domains.

Mapping linguistic tokens to the semantics specification is the next step in the approach. Token comprises of word/phrase from the corpus and the semantic attributes define in the previous step. The output of the approach is a set of UML models that will be subjected to review of human analyst. It is up to the analyst to corroborate the output of the approach and make an informed decision.

V. WORK PLAN

In order to accomplish this, we have defined a work programme. The work programme includes work on foundations, framework development and applications inform of case study evaluation.

- **Conceptualization** – this involves designing of the proposed framework and testing of the existing algorithms.
- **Tool development** – implementation of the tool support base on the proposed framework. Highlight of the development technique languages.
- **Applications** – this has to do with the evaluation through industrial case studies to establish the applicability of the proposed framework and generate useful feedbacks.

VI. CONCLUSION

This paper proposes a new framework for automated requirements engineering using natural language processing and artificial intelligence techniques. The proposed framework herein will be implemented and evaluated using at least two institutions/organizations as case studies. This is to establish the applicability of the framework in order to serve as one of the several empirical evidences/sources for academic and corporate discourse and application.

ACKNOWLEDGMENT

This work is funded by the Petroleum Technology Development Fund (PTDF) of the Federal Government of Nigeria.

 ISBN: 978-1-61208-827-3

REFERENCES

[1] H. F. Hofmann and F. Lehner, "Requirements engineering as a success factor in software projects," *IEEE Software*, vol. 18, no. 4, pp. 58–66, 2001, doi: 10.1109/MS.2001.936219.

[2] T. Shah and S. Patel, "A Novel Approach for Specifying Functional and Non-functional Requirements Using RDS (Requirement Description Schema)," *Procedia Computer Science*, vol. 79, pp. 852–860, 2016, doi: 10.1016/j.procs.2016.03.083.

[3] H. Meth, M. Brhel, and A. Maedche, "The state of the art in automated requirements elicitation," *Information and Software Technology*, vol. 55, no. 10, pp. 1695–1709, Oct. 2013, doi: 10.1016/j.infsof.2013.03.008.

[4] D. K. Deeptimahanti and M. A. Babar, "An Automated Tool for Generating UML Models from Natural Language Requirements," in *2009 IEEE/ACM International Conference on Automated Software Engineering*, Auckland, New Zealand, Nov. 2009, pp. 680–682, doi: 10.1109/ASE.2009.48.

[5] E. Parra, C. Dimou, J. Llorens, V. Moreno, and A. Fraga, "A methodology for the classification of quality of requirements using machine learning techniques," *Information and Software Technology*, vol. 67, pp. 180–195, Nov. 2015, doi: 10.1016/j.infsof.2015.07.006.

[6] J. Cleland-Huang, R. Settimi, X. Zou, and P. Solc, "Automated classification of non-functional requirements," *Requirements Eng*, vol. 12, no. 2, pp. 103–120, May 2007, doi: 10.1007/s00766-007-0045-1.

[7] N. A. Moketar, M. Kamalrudin, S. Sidek, M. Robinson, and J. Grundy, "TestMEReq: generating abstract tests for requirements validation," in *Proceedings of the 3rd International Workshop on Software Engineering Research and Industrial Practice - SER&IP '16*, Austin, Texas, 2016, pp. 39–45, doi: 10.1145/2897022.2897031.

[8] W. Miao *et al.*, "Automated Requirements Validation for ATP Software via Specification Review and Testing," in *Formal Methods and Software Engineering*, vol. 10009, K. Ogata, M. Lawford, and S. Liu, Eds. Cham: Springer International Publishing, 2016, pp. 26–40.

[9] R. Sharma and K. K. Biswas, "A Semi-automated Approach towards Handling Inconsistencies in Software Requirements," in *Evaluation of Novel Approaches to Software Engineering*, vol. 410, L. A. Maciaszek and J. Filipe, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 142–156.

[10] A. Rago, C. Marcos, and J. A. Diaz-Pace, "Identifying duplicate functionality in textual use cases by aligning semantic actions," *Softw Syst Model*, vol. 15, no. 2, pp. 579–603, May 2016, doi: 10.1007/s10270-014-0431-3.

[11] Y. Li, T. Yue, S. Ali, and L. Zhang, "Enabling automated requirements reuse and configuration," *Softw Syst Model*, vol. 18, no. 3, pp. 2177–2211, Jun. 2019, doi: 10.1007/s10270-017-0641-6.

[12] D. Ko, S. Kim, and S. Park, "Automatic recommendation to omitted steps in use case specification," *Requirements Eng*, vol. 24, no. 4, pp. 431–458, Dec. 2019, doi: 10.1007/s00766-018-0288-z.

[13] N. Yusop, M. Kamalrudin, S. Sidek, and J. Grundy, "Automated Support to Capture and Validate Security Requirements for Mobile Apps," in *Requirements Engineering Toward Sustainable World*, vol. 671, S.-W. Lee and T. Nakatani, Eds. Singapore: Springer Singapore, 2016, pp. 97–112.

[14] S. M. Edgar, B. S. Oscar, and S. A. Alexei, "Knowledge meaning and management in requirements engineering," *International Journal of Information Management*, vol. 37, no. 3, pp. 155–161, Jun. 2017, doi: 10.1016/j.ijinfomgt.2017.01.005.

[15] H. Meth, A. Maedche, and M. Einoeder, "Is Knowledge Power? The Role of Knowledge in Automated Requirements Elicitation," in *CAiSE 2013, LNCS 7908*, Springer-Verlag Berlin Heidelberg, 2013, pp. 578–593.

[16] D. Berry, R. Gacitua, P. Sawyer, and S. F. Tjong, "The Case for Dumb Requirements Engineering Tools," in *REFSQ 2011. LNCS*, vol. 7195, Springer, Heidelberg, 2012, pp. 211–217.

[17] B. Aysolmaz, H. Leopold, H. A. Reijers, and O. Demirörs, "A semi-automated approach for generating natural language requirements documents based on business process models," *Information and Software Technology*, vol. 93, pp. 14–29, Jan. 2018, doi: 10.1016/j.infsof.2017.08.009.

[18] I. Summerville, *Software Engineering*, 9th ed. Person Education, Inc., 2011.

[19] N. A. Moketar, M. Kamalrudin, S. Sidek, M. Robinson, and J. Grundy, "An automated collaborative requirements engineering tool for better validation of requirements," in *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering - ASE 2016*, Singapore, Singapore, 2016, pp. 864–869, doi: 10.1145/2970276.2970295.

[20] Y. Li, E. Guzman, K. Tsiamoura, F. Schneider, and B. Bruegge, "Automated Requirements Extraction for Scientific Software," *Procedia Computer Science*, vol. 51, pp. 582–591, 2015, doi: 10.1016/j.procs.2015.05.326.

[21] M. Elallaoui, K. Nafil, and R. Touahni, "Automatic Transformation of User Stories into UML Use Case Diagrams using NLP Techniques," *Procedia Computer Science*, vol. 130, pp. 42–49, 2018, doi: 10.1016/j.procs.2018.04.010.

[22] V. B. R. Vidya Sagar and S. Abirami, "Conceptual modeling of natural language functional requirements," *Journal of Systems and Software*, vol. 88, pp. 25–41, Feb. 2014, doi: 10.1016/j.jss.2013.08.036.

[23] V. Antinyan and M. Staron, "Rendex: A method for automated reviews of textual requirements," *Journal of Systems and Software*, vol. 131, pp. 63–77, Sep. 2017, doi: 10.1016/j.jss.2017.05.079.

[24] V. Ambriola and V. Gervasi, "On the Systematic Analysis of Natural Language Requirements with CIRCE," *Autom Software Eng*, vol. 13, no. 1, pp. 107–167, Jan. 2006, doi: 10.1007/s10515-006-5468-2.

[25] R. Gacitua, P. Sawyer, and V. Gervasi, "Relevance-based abstraction identification: technique and evaluation," *Requirements Eng*, vol. 16, no. 3, pp. 251–265, Sep. 2011, doi: 10.1007/s00766-011-0122-3.

[26] Y. Alkhader, A. Hudaib, and B. Hammo, "Experimenting With Extracting Software Requirements Using NLP Approach," in *2006 International Conference on Information and Automation*, Colombo, Sri Lanka, Dec. 2006, pp. 349–354, doi: 10.1109/ICINFA.2006.374136.

[27] J. L. Cybulski and K. Reed, "Computer-assisted analysis and refinement of informal software requirements documents," in *Proceedings 1998 Asia Pacific Software Engineering Conference (Cat. No.98EX240)*, Taipei, Taiwan, 1998, pp. 128–135, doi: 10.1109/APSEC.1998.733606.

[28] J. Natt och Dag, T. Thelin, and B. Regnell, "An experiment on linguistic tool support for consolidation of requirements from multiple sources in market-driven product development," *Empir Software Eng*, vol. 11, no. 2, pp. 303–329, Jun. 2006, doi: 10.1007/s10664-006-6405-5.

[29] Q. A. Do, S. R. Chekuri, and T. Bhowmik, "Automated Support to Capture Creative Requirements via Requirements Reuse," in *Reuse in the Big Data Era*, vol. 11602, X. Peng, A. Ampatzoglou, and T. Bhowmik, Eds. Cham: Springer International Publishing, 2019, pp. 47–63.

[30] G. Lucassen, M. Robeer, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper, "Extracting conceptual models from user stories with Visual Narrator," *Requirements Eng*, vol. 22, no. 3, pp. 339–358, Sep. 2017, doi: 10.1007/s00766-017-0270-1.

[31] I. Reinhartz-Berger and M. Kemelman, "Extracting core requirements for software product lines," *Requirements Eng*, vol. 25, no. 1, pp. 47–65, Mar. 2020, doi: 10.1007/s00766-018-0307-0.

[32] A. Al-Hroob, A. T. Imam, and R. Al-Heisa, "The use of artificial neural networks for extracting actions and actors from requirements document," *Information and Software Technology*, vol. 101, pp. 1–15, Sep. 2018, doi: 10.1016/j.infsof.2018.04.010.

[33] N. Kiyavitskaya and N. Zannone, "Requirements model generation to support requirements elicitation: the Secure Tropos experience," *Autom Softw Eng*, vol. 15, no. 2, pp. 149–173, Jun. 2008, doi: 10.1007/s10515-008-0028-6.

 ISBN: 978-1-61208-827-3