Text-To-Model (TeToMo) Transformation Framework to Support Requirements Analysis and Modeling

Gayane Sedrakyan[1,2], Asad Abdi[1], Stéphanie M. Van Den Berg[2], Bernard Veldkamp[2] and Jos Van Hillegersberg[1]

[1]*Faculty of Behavioral, Management and Social Sciences, Section Industrial Engineering and Business Information Systems (IEBIS), University of Twente, Enschede, The Netherlands*

[2]*Faculty of Behavioral, Management and Social Sciences, Section Cognition, Data & Education (CODE), University of Twente, Enschede, The Netherlands*

{g.sedrakyan, s.abdiesfandani, stephanie.vandenberg, b.p.veldkamp, J.vanHillegersberg}@ utwente.nl

Keywords: Requirements Engineering, Requirements Analysis, Conceptual Modeling, Text Mining, Natural Language Processing, Requirement Analysis Automation, Model Generation.

Abstract: Requirements analysis and modeling is a challenging task involving complex knowledge of the domain to be engineered, modeling notation, modelling knowledge, etc. When constructing architectural artefacts experts rely largely on the tacit knowledge that they have built based on previous experiences. Such implicit knowledge is difficult to teach to novices, and the cost of the gap between classroom knowledge and real business situations is thus reflected in further needs for post-graduate extensive trainings for novice and junior analysts. This research aims to explore the state-of-the art natural language processing techniques that can be adopted in the domain of requirements engineering to assist novices in their task of knowledge construction when learning requirements analysis and modeling. The outcome includes a method called Text-To-Model (TeToMo) that combines the state-of-the-art natural language processing approaches and techniques for identifying potential architecture element candidates out of textual descriptions (*business requirements*). A subsequent prototype is implemented that can assist a knowledge construction process through (semi-) automatic generation and validation of Unified Modeling Lnaguage (UML) models. In addition, to the best of our knowledge, a method that integrates machine learning based method has not been thoroughly studied for solving requirements analysis and modeling problem. The results of this study suggest that integrating machine learning methods, word embedding, heuristic rules, statistical and linguistic knowledge can result in increased number of automated detection of model constructs and thus also better semantic quality of outcome models.

1 INTRODUCTION

Our current ICT-driven world of economies depends hugely on information systems. A tiny failure in an interconnected web of information systems of various organizations may result in significant and sometimes disastrous consequences. The quality of information systems and thus the process of designing and producing high quality systems becomes critical. The process of designing information systems is a highly complex and challenging task that requires rigorous analytical skills and experience. The first step in this process includes externalizing business requirements into formal model representations that serve the first artefact for formal evaluation of quality. With the growing importance of compliance between business strategy and ICT realizations, as well as popularization of low/no code platforms that allow business and citizen developers to design and produce business applications, conceptual models gain relevance. Conceptual models use highly abstract representations and can significantly reduce the complexity of a problem domain, thus making it easier to integrate business domain and ICT expertise in the system design process. Conceptual models also contain the critical information for designing and applying effective organizational strategies and a necessary foundation for constructing an organization's information system. Important key factors affecting the quality of a conceptual model are knowledge of modeling concepts, of the modeling language and of the domain to be modelled. Teaching

Sedrakyan, G., Abdi, A., Van Den Berg, S., Veldkamp, B. and Van Hillegersberg, J.
Text-To-Model (TeToMo) Transformation Framework to Support Requirements Analysis and Modeling.
DOI: 10.5220/0010771600003119
In *Proceedings of the 10th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2022)*, pages 129-136
ISBN: 978-989-758-550-0; ISSN: 2184-4348

such knowledge and skills to novice modelers is a challenging task considering that system analysis is by nature an inexact skill (Sedrakyan, 2016). Good modelers rely mainly on their personal experience, and the tacit knowledge that they have developed over time, which is difficult to transfer to junior modelers. Transferring the academic knowledge and skills to real world businesses is yet another concern as the classroom and real-world situations are not identical. As stated by Schenk, Vitalari, and Davis (1998), in their early careers novice modelers produce incomplete, inaccurate, ambiguous, and/or incorrect information requirements. Several reasons make conceptual modeling skills very difficult to teach. For instance, studies on comparing model quality checking approaches of novices and experts indicate the poorly adapted cognitive schemata of novice modelers to identify relevant triggers for verifying the quality of models. Previous research on observing the differences in modeling process indicates the linear problem-solving pattern of novices focusing on one task at a time vis-a-vis experts' frequent switches between modeling activities and simultaneous cross-validation cycles (80% of their design activity) (Wang & Brooks, 2007). Tools that assist a modeling and verification process include techniques that provide support at the level of modeling outcome, such simulation for verifying and validating the semantic quality of models as well as automated feedback facilitating the interpretation of the results and/or model verification. However, there is little support in the process of making design choices when transferring requirements into models, such as the choice of candidate components and their relationships in a model. Our research aims to explore state-of-the-art text mining techniques and capabilities in supporting requirements analysis and modeling task for novices. We start with mapping existing text mining concepts and techniques with the modeling technique (*such as diagram type*) and the constructs (*e.g. elements supported by the diagram type*) to come up with a framework that can guide the design and development of text-to-model (semi-) automated generation instrument. Testing the capabilities with respect to the most standard modeling techniques used in designing information systems constitutes a future research direction, however initial results (*as a demo*) from a pilot study is included which compares model constructs extracted by an experienced human modeler and those produced by using different text mining approaches. The remainder of the paper is structured as follows. Section 2 describes the research methodology used for the work. Section 3 gives an overview of related work. Section 4 describes the proposed method for text-to-model construction support. Our system evaluation is explained in Section 5. Finally, section 6 concludes the work with a discussion of results as well as proposing some future research directions.

2 RESEARCH METHOD

Theories are based on a systematic view of phenomena. Kerlinger (1979) proposes a method based on specifying relations among variables that use a set of interrelated constructs, variables, definitions, and propositions. Conceptualizations in the form of a framework guide research by providing a visual representation of theoretical constructs, variables of interest, and their relationships as suggested by Creswell (1994). Based on such frameworks, concrete applications can be developed Morgan (2018). In this paper, we aim to derive a framework based on earlier literature and previous empirical research, to guide the design and development of text-to-model transformation application to support (semi-)automatic requirements analysis and conceptual modeling process. This is achieved by conducting a literature study of existing text mining and/or machine learning techniques and further mapping these techniques with corresponding concepts in a modeling process, such as diagram types used for conceptual modeling task and their corresponding constructs such as class, attribute, association.

3 LITERATURE REVIEW

3.1 Teaching and Learning Context

In a teaching context, model comprehension difficulties, in addition to the lack of modeling knowledge, are also associated with the insufficient level of experience of novices and as a result their limited cognitive resources to identify relevant triggers for model verification. Unified Modeling Language (UML) is a standard language that is used to document and model business requirements. According to a complexity analysis by Siau and Cao (2001) UML (Unified Modeling Language) class diagram ranks the highest in complexity among the structural diagrams followed by state chart among the dynamic diagrams because of their high cognitive and structural complexity (Cruz-Lemus, Maes,

Genero, Poels, & Piattini, 2010). In a Delphi study by Erickson and Siau (2007) identifying the kernel of "essential" UML class diagram use cases, sequence and state chart are found to have the highest usability ranks by practitioners and educators from software industry and academic field. Furthermore, these are also among the top used diagrams present in the context of educational material such as books, tools, courses and tutorials, with percentages of 100% (class diagram, use case, sequence) and over 96% (State chart), while also being among the top diagramming techniques that support conceptual modelling.

3.2 Text Mining Techniques Applied for Requirements Engineering

Requirements analysis (RA) has a key role in the process of development of information systems. It is a challenging task that requires complex knowledge and skills. The quality of a requirements analysis affects the quality of the information systems to be engineered. Nowadays models are largely used for constructing an organization's information system thus also serving the first artefact the quality of which can be formally tested. UML uses a graphical notation that can represent both the structure and business logic of the system to be engineered. Usually, requirements documents and domain descriptions are provided in natural language. Processing these documents using text mining applications (TMA) can help to extract information that may be useful in a learning context (e.g. by hinting candidate model constructs) to assist novices, or in the context of processing large requirement documents by experts (e.g. by identifying potential constructs that can be confirmed or discarded by a human modeler). For instance, TMA can assist transforming an unstructured data set into a structured format or a medium that can be used to generate business model constructs. Finally, a formal diagram can be drawn for the system from business requirements automatically. Earlier research proposed various methods to draw UML diagrams from using text mining approach. We will briefly discuss some of those we consider to be of particular interest.

Montes, Pacheco, Estrada, and Pastor (2008) presented a natural language processing based method to generate UML diagrams using a plain text as an input. The method analyses the given script/text to extract relevant information, based on which UML diagrams are drawn. The process of creating, arranging, labeling and finalizing the UML diagrams is performed using the following steps: 1. Text input acquisition, to read and obtain input text scenario, 2. Syntactic Analysist, to categorize words into various classes as verbs, helping verbs, nouns, pronouns, adjectives, prepositions, conjunctions, etc. 3. Text understanding to infer the meanings of the given text by using semantic rules (Malaisé, Zweigenbaum, & Bachimont, 2005), 4. Knowledge extraction, which extracts required data attributes using a set of rules (Van Rijsbergen, 1977), 5. UML diagram generation uses UML notation symbols to draw a UML diagram. Shahzadi, Ahmad, Fatima, Sarwar, and Mahmood (2013) proposed a method to identify domain entities and their relationships from text documents that can be transformed into a UML diagram. The method performs a linguists processing on a given text using open source tool named GATE (Cunningham et al., 2009) It allows marking entities and relationships between entities. Their system includes the following steps: 1. document acquisition, 2. document processing, 3. XML modeling. Document Acquisition step obtains an input from a textual document. Document processing step applies linguistic processing (e.g., sentence splitter, tokenizer, part-of-speech tagger). XML modeling is used to convert a textual data into a formal data-model. Harmain and Gaizauskas (2000) developed a method which produces an object-oriented model from textual documents. Montes et al. (2008) present a method of generating an object-oriented conceptual model (e.g., UML class diagrams) from natural language text. Hasegawa, Kitamura, Kaiya, and Saeki (2009) also introduced a tool that extracts requirements models from natural language texts. Gelhausen, Derre, and Geiß (2008) and Gelhausen and Tichy (2007) presented a method to create UML domain models directly from a textual document. The authors employ a graph technique as an intermediate representation of a text. The nodes in the graph represent sentences and words. Edges indicates thematic roles and are the core component of the method as they represent the semantic information in the text. Graph transformation rules are then used to build a UML representation. Mala and Uma (2006) use an NLP pipeline to create a model without the intervention of a domain expert. The authors claim that the yielded results are at least as good as or exceeding human-made class diagrams. Bajwa and Choudhary (2006) extract nouns and verb combinations from input texts and map the nouns and verbs to UML class elements and relations, respectively. However, to our knowledge, little is known about how these text mining techniques and applications can support the task of deriving business requirements and automatic generation of formalized requirements documents, such as UML models, from

a theoretical point of view that builds on evidences from prior studies in the domain. In addition, methods that integrate machine learning-based method, word embedding, heuristic rules, statistical and linguistic knowledge also included in this method, have not been thoroughly studied for solving requirements analysis and modeling problem. This research aims to propose a method that 1. builds on the analysis and summary of the literature on the applications of text mining in the domain of requirements analysis and modeling and 2. generalizes to the next level by introducing the gaps in this domain for future research needs. We argue that, while the application of text mining in the business domain is still not mature enough to produce requirements documents and models that are of high semantic quality (such as accurate, complete, etc.) there is a good potential for these techniques to assist a human analyst/modeler to facilitate some steps in the process, but also assist a learning/training process of a novice analyst. The latter can be achieved not only by automated discovery and proposal of candidate constructs such as entities, attributes, (*mandatory and optional*) relationships, activities, sequences, etc., but also learning from expert analysis and modeling process/behavior, or a business domain document generate recommendations and feedback for a learning context.

Summing up, the problem in the domain discussed in the literature so far deals with the fact that there is a need for supporting techniques that can assist the analysts for routine tasks of systematically producing a model appropriately representing the expected structure and functionality of a system with less effort. To achieve this, most of the earlier works used shallow linguistic techniques, while the method proposed in this work aims to in addition solve and address some of the complex linguistic problems applied within the domain of business requirements.

The contribution of this work can be thus addressed as follows:

1) The work is the first attempt of integrating a machine learning-based method, word embedding, heuristic rules, statistical and linguistic knowledge, which has not been thoroughly studied for solving software requirements analysis.

2) The TeToMo based implementation employs several meaningful resource-information latent in a sentence to a) learn a better sentence representation; b) create augmented vector; c) obtain significant performance. A hybrid vector is created to represent each sentence using the statistical, linguistic knowledge-based and word embedding-based feature vectors.

3) The TeToMo based implementation generates an XMI file (Specification, 2006) as output (currently limited to modeling constructs of class and state chart models) that is generic enough to allow to be imported and visualized in any UML modeling tool. Furthermore, a human analyst can interfere in the generation process and/or further refine and extend them.

4) Finally, we conduct experiments for performance evaluation and comparison. We report our results on the benchmark dataset. In the other words, we measure the performance of the TeToMo against human judgment to confirm the suitability of the proposed method for analysing software requirements documents.

4 TeToMo FRAMEWORK

Models of a system-to-be often combine structural and behavioral views. This can be achieved by combining different diagrams such as class diagram, process flows represented as business process models, flowcharts, activity diagrams, state chart, etc. Potential elements that a modeler may need to tag/derive from business requirements during early stages of analysis (which we will further refer to as ***model constructs***) include for instance:

- Entities (class diagram).
- Attributes (class diagram).
- Relationships/associations between entities (class diagram).
- Type of relationship/association, e.g. mandatory vs. optional (class diagram).
    - wording like may, can, … show modality, wording like must, should,… may denote mandatory relationship.
- Events (activity diagram, state chart).
- Sequences of events (activity diagram, state chart).
    - event A may occur after or before event B, etc., event X is a starting event, event Y is an ending event.
- States (state chart) that are usually verbs in passive form (e.g. approved, blocked, ended, started, etc.).

The architecture of TeToMo based implementation of model extraction is displayed in Figure 1. It illustrates the steps and functionality of the system to transform user requirements such as user stories into UML diagrams. There are six main modules in the TeToMo system. *It is worth noting, because of space limitation, we explain each of them in summary:* Input document includes the document with business requirements for processing. The Natural Language Analysis (NLA)

module is applied to the document and the processed information is sent to the Relationship, Object, Method, Attribute (ROMA) module to extract modeling constructs such as the attributes, object/concepts and relationships between objects using both NLP module, heuristic rules and machine learning-based method. The findings are further proposed to the user as candidate model constructs through the Interference Human Interface which can be used to approve, discard, make duplicate constructs based on e.g. synonyms, etc. Model Generator Module is used to compose a model and subsequent structural and dynamic views such as a class diagram and state charts. XMI format is employed to store the information of finalized model constructs and relationships. Finally an XMI file can be imported to a UML modeling tool to draw the required diagram. Detailed explanation of these modules is as follows:

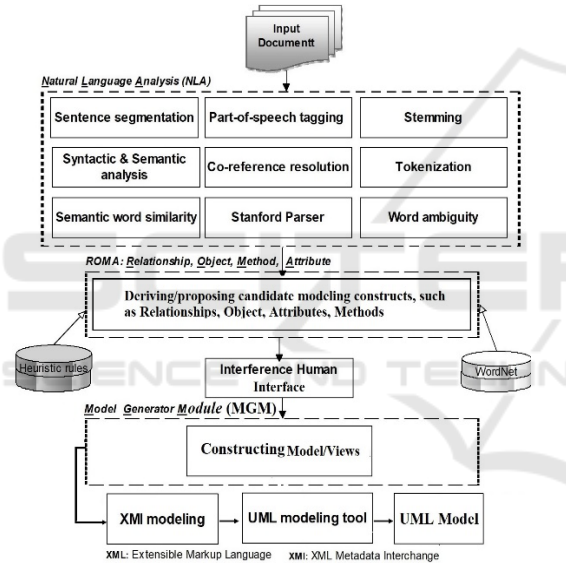

Figure 1: The Architecture of the method.

Input Document: The goal of this module is to collect a textual document (business requirements articulated in natural language either in unstructured or structured medium format) as an input to the system.

Natural Language Analysis (NLA) Module: This module uses basic linguistic functions to analyse the text syntactically and semantically, and stores all the intermediate processing results for further analysis. This stage includes the following tasks: The raw text is first processed. Then, the document is decomposed into several paragraphs. Next, the paragraphs are further decomposed into sentences. Secondly, the tokenization, a basic approach of the text pre-processing, splits the sentences into words. We also employ the stemming procedure to reduce a word to its root form. Part of Speech (POS) tagging is also used to classify the words of text on the basis of part of speech category (‘*noun*’, ‘*verbs*’, ‘*adverb*’, ‘*adjectives*’) they belong. The POS tagging provides useful lexical information.

The syntactic analysis is used to identify subject, verbs, objects, adverbs, adjectives and various other complements. The NLA also uses other basic linguistic functions such as co-reference resolution, stemming process, semantic role labeling (active & passive sentences), semantic word similarity (word embedding & statistical method), WordNet dictionary to enhance the semantic analysis. The results generated by the NLA is passed to the ROMA module (see Figure 1).

Sentences segmentation function is used to split a paragraph into sentences. *Stemming* technique is used to reduce word to its stem form (e.g., ‘*went*’---> ‘*go*’). *Stanford Parser* is used to produce a parse tree for each sentence to identify the ‘phrases’, the object or subject of a verb.

WordNet is an English language lexical database. It can be employed to obtain semantically similar terms and acquire synonyms. The synonyms are used to extract words that are semantically related to each other (e.g., term frequency (*TF*) method uses synonym to find a word with high frequency in a text file). *Co-reference Resolution* process is used to determine linguistic expressions that refer to the same entity in a text. This is useful to link pronominal references (e.g., she, it) to earlier referent in the text.

ROMA module has two sub-modules for identification of model constructs.: *a)* Objects, attributes and methods identification: The main task of the current sub-module is extracting the important words that can be considered as a model construct such as entity/concept, corresponding attributes related to each concept, the method/operations and involvement of entity instances using the calculation of various measure parameters of the words (e.g., *TF* (*term frequency*), *TF*IDF* (*term frequency × inverse document frequency*), *Entropy, C-value (Hasegawa et al., 2009))*, Heuristic rules and the basic linguistic functions (e.g., *parser*). *b)* Relationship identification: A machine learning-based method employs word embedding method, various heuristic rules (Narawita, 2017), specific words, various measure parameters, statistical and linguistic knowledge as input features to identify different types of relationships.

Interference Human Interface: TeToMo provides an interactive user interface that allows an end-user to manage and analyze business requirements. S/he can choose to interfere in the process of initial tagging as

well as the selection of the final model constructs out of those proposed by the ROMA module. The system presents a list of candidate constructs a user can make a decision whether or not to include or exclude a particular item (e.g., '*redundant classes*', '*irrelevant classes'*).

Model Generator Module: This module generates different UML diagrams using pre-processed requirements. It uses the extracted Information of 'ROMA' module and Interference Human interface to extract model views such as class diagram and state chart diagram. Algorithms are used to identify and determine the class and State chart models elements (e.g., concepts/objects/entities, attributes, signal/operation/method, associations, relationships, Is-a relationship (*generalization*), Has-a (aggregation), multiplicities on relationships among the objects, states, events, sequences).

Heuristic rules include a set of rules to identify objects, attributes, method, the multiplicity of roles in associations and relationship between the objects. To do this, we collected several heuristic rules from previous studies (Deeptimahanti & Sanyal, 2011) such as syntactic reconstruction rules to split a complex sentence into simple sentences to extract all possible information from the requirements document. *An XMI file* is produced as the final output of the TeToMo. This file includes the information about the identified concept/entity/object, their attributes, and the relationships among them, their typology and relations linked to UML notation at meta-data level. It is used to visualize generated models in any UML modeling tool which supports the XMI import facility or XML compatible.

5 SYSTEM EVALUATION

The evaluation of TeToMo will be conducted through the performance evaluation by comparing the output of the TeToMo with the model generated manually (*by an expert*). For this purpose different case studies from different domains have been used. The aim of the system evaluation is to assess the extracted constructs and model views with respect to its semantic quality measured by semantic conformance with the business requirements, completeness, and accuracy. Furthermore, this section aims to answer the following question: how close the model generated by the TeToMo is to the one produced by an expert analyst.

Standard Data Set — to analyze the performance of TeToMo, we also need a gold standard model (*a set of all correct results*). For this purpose, an expert with sufficient skill, experience and domain knowledge is asked to produce the corresponding UML diagram for each use case. The produced models are then compared with the model generated by TeToMo at the level of model constructs such as attribute, relationship, object/concept. In other words, we compare which parts of models generated by the expert and TeToMo are the similar, complementary or conflicting.

Evaluation Metrics — we use three standard metrics to evaluate the performance of the TeToMo. These metrics known as Recall (R), Precision (P) and *F*_measure. Precision is a set of selected items that are true, while recall is a set of correct items that are selected. In this study, the model elements (*e.g., attribute, relationship, object/concept*) identified by a human refers to a set of ideal items, and the model elements identified by the TeToMo refers to a set of system items. In other words, precision is used to assess the fraction of the system items that the TeToMo correctly identified and recall is used to assess the fraction of the ideal items that the algorithm identified. The precision is calculated using Eq. (1): the division of identified model elements by TeToMo and human expert over the number of model elements identified by TeToMo only. The recall is calculated using Eq. (2): the division of identified model elements by TeToMo and human expert intersection over the number of model elements identified by a human expert.

$$Precision = \frac{M}{M+N} \quad (1)$$

$$Recall = \frac{M}{M+K} \quad (2)$$

Where, M = The number of model elements identified by TeToMo and Human expert. N = The number of model elements identified by TeToMo only. K = The number of model elements identified by Human expert only. Furthermore, the *F*-measure is used to merge both precision and recall. It is computed as follows:

$$F\_\text{measure} = \frac{2 \times P \times R}{P+R} \quad (3)$$

Sample Case Study: Figure 2 presents a conceptual model of simplified a ticket sale system generated by an expert. *"A company sells airline tickets, for which customers can make reservations either on a company's online booking system or by calling a phone operator. To meet customer demand the company uses a flexible pricing policy based on which a seat prices is decided during the reservation by a manager. Furthermore, the price for a seat is not*

fixed and can change depending on the changes of several conditions, e.g. after a reservation is cancelled, and there are not many seats available for the flight while having a high demand (e.g. due to popular vacation destination), the price for the seat can increase".

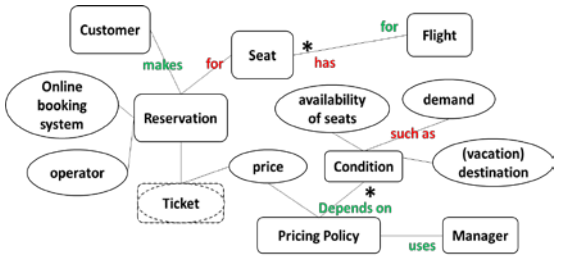

Figure 2: Sample intermediate diagram produced by TeToMo (missing to detect elements noted in red).

Table 1: Precision, Recall and F-score results.

	Class name *Extracted By*		The value of			Evaluation metrics		
Case study	Human	TeToMo	M	N	K	P	R	F_measure
1	13	16	13	16	13	0.45	0.50	0.47
2	14	18	14	18	14	0.44	0.50	0.47
…	…	…	…	…	…	…	…	…
n	12	9	10	9	12	0.53	0.45	0.49
					Average:	0.47	0.48	0.47

The model shows classes/concepts (e.g., 'Customer', 'Seat', 'Reservation', etc.). These concepts are linked to each other through relationships. In addition, each class includes several attributes, methods/operations. On the other hand, a TeToMo produced sample of the corresponding conceptual model from the above problem statement produced the same output (with the missed elements compared to human modeler noted in red)**.** To evaluate the method, we measure the performance of the TeToMo against human judgment on the level of the extracted model elements. For instance, Table 1 presents the three main columns: number of concepts/class name derived from Figures 2 (e.g., the expert extracted 7 concepts and the TeToMo extracted 6 concepts). To calculate the P, R, and F-measure, we determine the values of M, N, K. Then, the equations of (1), (2) and (3) are applied to get the value of each evaluation metric. The averages of precision, recall and F-measure are considered as a performance of the TeToMo. We also consider using the same format presented in Table 1 to evaluate the detection of other elements such as attributes, methods, relationships. In this case, the column name "*Concepts/Class Name*" is replaced by the "*Attribute*", "*Method*" or "*Relationship*", etc.

6 CONCLUSION

The work proposes a method that can be used to design and develop a text mining based support for requirements analysis and modeling process. A prototype solution was designed for a conceptual modeling case using well-known standard diagrams for designing structural and behavioral aspects of a system. In this paper, we present a novel method that integrates machine learning based-method, word embedding, heuristic rules, statistical and linguistic knowledge to solve requirements analysis and modeling problem (*called TeToMo*).

The effectiveness will be evaluated by comparing model solutions designed by human modeler and those constructed with the help of the prototype, e.g. based on the candidate constructs proposed by text mining approach. In addition, we also aim to compare the performance of TeToMo with those of the other existing and well-known methods used for generating UML diagram such as (Landhäußer, Körner, & Tichy, 2014). While this work outlines some findings on the method performance from a pilot study, evaluating the effectiveness of the method with an enhanced prototype and replication studies using a larger corpus constitutes a further research direction. Bi-directional verification of requirements from models to text is yet another direction for further work. Finally, discourse analysis module can be added to the NLA to allow determining contextual information. Addressing this points will lead to improved versions of the TeToMo framework and applications. While in this work the framework is limited to two specific types of diagrams, expanding the scope to experiment with more UML diagram types will be a step towards a more generic approach.

REFERENCES

Bajwa, Imran Sarwar, & Choudhary, M Abbas. (2006). *Natural language processing based automated system for uml diagrams generation.* Paper presented at the The 18th Saudi National Computer Conf. on computer science (NCC18). Riyadh, Saudi Arabia: The Saudi Computer Society (SCS).

Creswell, John W. (1994). Research design: Qualitative and quantitative approach. *London: Publications.*

Cruz-Lemus, José A, Maes, Ann, Genero, Marcela, Poels, Geert, & Piattini, Mario. (2010). The impact of structural complexity on the understandability of UML statechart diagrams. *Information Sciences, 180*(11), 2209-2220.

Cunningham, Hamish, Maynard, Diana, Bontcheva, Kalina, Tablan, Valentin, Ursu, Cristian, Dimitrov, Marin, . . . Li, Yaoyong. (2009). *Developing Language Processing Components with GATE Version 5:(a User Guide)*: University of Sheffield.

Deeptimahanti, Deva Kumar, & Sanyal, Ratna. (2011). *Semi-automatic generation of UML models from natural language requirements.* Paper presented at the Proceedings of the 4th India Software Engineering Conference.

Erickson, John, & Siau, Keng. (2007). *Can UML be simplified? Practitioner use of UML in separate domains.* Paper presented at the proceedings EMMSAD.

Gelhausen, Tom, Derre, Bugra, & Geiß, Rubino. (2008). *Customizing grgen. net for model transformation.* Paper presented at the Proceedings of the third international workshop on Graph and model transformations.

Gelhausen, Tom, & Tichy, Walter F. (2007). *Thematic role based generation of UML models from real world requirements.* Paper presented at the International Conference on Semantic Computing (ICSC 2007).

Harmain, Harmain Mohamed, & Gaizauskas, R. (2000). *CM-Builder: an automated NL-based CASE tool.* Paper presented at the Proceedings ASE 2000. Fifteenth IEEE International Conference on Automated Software Engineering.

Hasegawa, Ryo, Kitamura, Motohiro, Kaiya, Haruhiko, & Saeki, Motoshi. (2009). *Extracting conceptual graphs from Japanese documents for software requirements modeling.* Paper presented at the Proceedings of the Sixth Asia-Pacific Conference on Conceptual Modeling-Volume 96.

Kerlinger, Fred N. (1979). *Behavioral research: A conceptual approach*: Holt, Rinehart and Winston New York.

Landhäußer, Mathias, Körner, Sven J, & Tichy, Walter F. (2014). From requirements to UML models and back: how automatic processing of text can support requirements engineering. *Software Quality Journal, 22*(1), 121-149.

Mala, GS Anandha, & Uma, GV. (2006). *Automatic construction of object oriented design models [UML diagrams] from natural language requirements specification.* Paper presented at the Pacific Rim International Conference on Artificial Intelligence.

Malaisé, Véronique, Zweigenbaum, Pierre, & Bachimont, Bruno. (2005). Mining defining contexts to help structuring differential ontologies. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication, 11*(1), 21-53.

Montes, Azucena, Pacheco, Hasdai, Estrada, Hugo, & Pastor, Oscar. (2008). *Conceptual model generation from requirements model: A natural language processing approach.* Paper presented at the International Conference on Application of Natural Language to Information Systems.

Morgan, David L. (2018). Themes, theories, and models. *Qualitative health research, 28*(3), 339-345.

Narawita, Chamitha Ramal. (2017). UML Generator-Use Case and Class Diagram Generation from Text Requirements. *ICTer, 10*(1).

Schenk, Karen D, Vitalari, Nicholas P, & Davis, K Shannon. (1998). Differences between novice and expert systems analysts: What do we know and what do we do? *Journal of management information systems, 15*(1), 9-50.

Sedrakyan, Gayane. (2016). Process-oriented feedback perspectives based on feedback-enabled simulation and learning process data analytics, PhD Thesis, KU Leuven.

Shahzadi, Iram, Ahmad, Qanita, Fatima, Kiran, Sarwar, Imran, & Mahmood, Waqar. (2013). UMagic! THE UML modeler for text documents. *International Journal of Computer Theory and Engineering, 5*(1), 166.

Siau, Keng, & Cao, Qing. (2001). Unified modeling language: A complexity analysis. *Journal of Database Management (JDM), 12*(1), 26-34.

Specification, OMG Business Process Modeling Notation. (2006). Object management group. *Needham, MA, USA, 2*(2).

Van Rijsbergen, Cornelis Joost. (1977). A theoretical basis for the use of co ‐ occurrence data in information retrieval. *Journal of documentation*.

Wang, Wang, & Brooks, Roger J. (2007). *Empirical investigations of conceptual modeling and the modeling process.* Paper presented at the 2007 Winter Simulation Conference.