A Semi–automated Approach for Generating Sequence Diagrams from Arabic User Requirements Using a Natural Language Processing Tool

Nermeen Alami
Palestine Polytechnic University
Deanship of Graduate Studies
and Scientific Research
Hebron, Palestine

Nabil Arman
Department of Mathematics
and Computer Science
Palestine Polytechnic University
Hebron, Palestine

Faisal Khamyseh
Department of Mathematics
and Computer Science
Palestine Polytechnic University
Hebron, Palestine

Abstract—a sequence diagram is one of the UML models that are usually used within the analysis phase in software system development. Since generating such sequence diagrams is done usually in a manual way, automated or semi-automated support will be appreciated and will provide important and practical help. In this paper, we propose a new semi-automated approach for generating sequence diagrams from user requirements written in Arabic. In this novel approach the user Arabic requirements are parsed using a natural language processing tool to generate the part of speech tags of the parsed user requirements. A set of proposed heuristics are to be applied to obtain the sequence diagram components; objects, messages and work flows transitions (messages). The generated sequence diagram is to be represented using XMI to be drawn using sequence diagrams drawing tools. Using three different case studies as a bench mark from Isra Computer and Programming Company, the proposed approach will be evaluated in terms of correctness and completeness of participants and messages exchanged between them.

Keywords—Unified Modeling Language (UML), Automated Software Engineering, Sequence Diagram

I. INTRODUCTION

The sequence diagram shows how processes in software sys-tem interact with each other based on time. In analysis phase it is used to illustrate the objects that participate in the use case and the messages passing between them over time while in design phase the sequence diagram is used to distribute the use case behavior to classes [1], [2].

This paper addresses the problem of generating sequence diagram from Arabic user requirements, written in Arabic, in a semi-automated approach using a natural language processing tool namely MADA+TOKAN. All UML models are usually produced based on user requirements. The process of transforming the user requirements into the UML diagrams is normally done by human analysts which is time and money consuming and also an error-prone process because the user requirements are usually written in natural language. The human analyst may make mistakes during reading a large number of natural language user requirements which may produce an incorrect model. In addition, if a change is needed to be applied to the model then a lot of effort, money and time will be wasted during the modification process in order to accommodate the needed changes. So, the need for an automated or semi-automated approach becomes urgent to save a lot of time, effort, and money [10].

The rest of the paper is organized as follows: the section of related works presents the literature studies and any related works; the section of constructing sequence diagrams describes the methodology used for generating the sequence diagram models from Arabic user requirements; the section of validation presents the validation and implementation of our proposed approach, and finally, the section of conclusion presents the main issues related to the proposed approach

II. RELATED WORKS

In general, the related literature studies about generating sequence diagrams can be mainly divided into two types; the full automation and semi-automation of sequence diagrams. In both types, as it has been stated in many related studies, the generation of sequence diagrams usually depends on some UML diagrams such as class diagrams and use case diagrams as a first step before generating the sequence diagrams, however there were no studies to generate sequence diagrams directly from Arabic user requirements or without using other UML diagrams as pre-step.

A. Semi-automated Methods for Generating Sequence Diagrams

Recent studies presented semi-automated approaches for generating sequence diagrams using use case or other UML diagrams [12], [4], [17], [6]. Thakur and Gupta presented a semi-automatic approach to translate the use case descriptions into sequence diagrams [7]. The study also presented a set of rules for writing and rewriting the descriptions of use case diagram that can be understood and helpful for both developers and experts which also can be then transformed and translated to build the sequence diagrams. B. Full-automated Methods for Generating Sequence Dia-grams

B. Full-automated Methods for Generating Sequence Diagrams

A method that uses the use case specifications (UCS) in generating a sequence diagram is presented by Mason and Supsrisupachai [8]. In this study the generation of sequence diagram was based on UCS written in Spanish language. Yue et al., proposed an approach to automatically generate the

sequence diagrams from use case specifications, UCSs is presented [18] in which the objects are identified using a set of heuristic rules.

C. Generating other UML Diagrams from User Requirements

A set of studies that are related to generating other diagrams from user requirements was published. A set of studies to propose algorithms for generating use case and activity diagrams from Arabic user requirements by [10], [11] were presented, in the first study a semi-automated algorithm for generating activity diagram from Arabic user requirements using MADA+TOKAN NLP tool. In which the elements of the activity diagram have been extracted from Arabic user requirements. The second study is also about generating use case diagram from user requirements written in Arabic in which a set of heuristic rules were proposed to obtain the use cases diagrams.

Another two important recent studies were about generating sequence diagrams from user stories written in English natural language [12], [13]. The first study used an algorithm worked by reading a text file of user stories and for each user story generated an XMI file which later on is transformed into sequence diagram using UML2 tool. The second study is about generating behavioral diagrams (sequence and activity diagrams) by transforming the statements of the requirements into a structured representation (intermediary structured using frames), in which those frames were translated into UML models. In this paper, the authors used grammatical knowledge patterns and lexical and syntactic analysis to analyze the requirements in order to get the frames for the corresponding requirements statements. By using the knowledge patterns in the resulted frames, the activity and sequence diagrams are generated. This study was presented using a set of performed case- studies.

As reported above, the generation of static and/or the dynamic models has been done using automatic and semi-automatic approaches. Most of the studies were for the purpose of deriving the static structure or class diagrams meanwhile the number of fully automated approaches was very few. Moreover, the sequence diagrams has been rarely generated in both types; the automated and the semi-automated. The reason behind that is that sequence diagrams differ from other UML diagrams, in which it cannot be mapped to graphical diagrams for sequence diagrams have the lifelines that are represented using vertical lines whereas the nodes in graphical diagrams are usually circles or boxes. Another reason for the scarcity of the sequence diagram research compared to the graphical diagram that has nodes is that the connection points in the graphical diagrams are usually placed on one of the sides of the node for incoming and outgoing connections whereas in sequence diagrams the messages are placed over the vertical line horizontally [10].

III. CONSTRUCTING SEQUENCE DIAGRAM MODELS

The main measure to find out the success of the software system is by measuring how much the output system meets the preset purpose and for what is intended to do. To have good results, we should have good requirements, and the good requirements should have a set of characteristics based on IEEE standards for Software Requirements Specifications, those characteristic require that user requirements should be correct, unambiguous, verifiable, traceable, complete and Consistent. It is assumed that the requirements are good in the sense implied by the IEEE good requirements assumptions [14], [15].

In this section the sequence diagram key parts are extracted from Arabic user requirements after using a natural language processing tool called MADA+TOKAN to split and tokenize Arabic user requirements texts. Once this is performed, a set of proposed heuristics are used to construct the sequence diagram model as presented in subsequent subsections. Finally the resulted sequence diagram is expressed in XMI to be drawn using UML drawing tools.

A. MADA+TOKAN

MADA+TOKAN is a Toolkit for Arabic Tokenization, Discretization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization [9]. MADA+TOKAN is a free tool, very customizable and versatile toolkit for NLP Arabic applications. It is for the purpose of extracting morphological and contextual information from the raw Arabic text in order to be used for other applications. It mainly consists of two main components: MADA and TOKAN. MADA is the service of giving new Arabic text by adding morphological and lexical information, while the TOKAN is the utility of generating segmentation (Tokenization) based on the information produced from the MADA process in order to identify the stem of the words. Having the two utilities together (MADA+TOKAN) provide a powerful tool for preprocessing for the applications of NLP such as Automatic Speech Recognition (ASR) [16], [17].

A set of user requirements cases of real system scenarios from ISRA SOFTWARE and PROGRAMMING COMPANY were written in Arabic and some of these requirements are used in our examples. ATM system example is presented below.

يقوم الزبون بإدخال البطاقة الى الصراف حيث يقوم الصراف بطلب الرقم السري من الزبون، يدخل الزبون الرقم السري الى الصراف ليقوم الصراف بإرسال المعلومات الى البنك للتحقق منها، يقوم البنك بإعادة النتيجة الى الصراف، ثم يقوم الصراف بعرض الخيارات للزبون، ويقوم الزبون بإدخال طلبه الى الصراف، يرسل البنك رسالة نجاح العملية الى الصراف، ويقوم الصراف بارسال اشعار تنفيذ العملية للزبون.

MADA+TOKAN results are two tags for each word as in

Table I and Table II; the first tag from Table I is the word type (verb, noun, punctuation, particle, etc.) while the second tag from Table II which is the word parsing (verb, subject, object, etc.). These tags based on the developed heuristics are used in determining the participants and messages of the sequence diagram as described below.

Table I: MADA+TOKAN Word Grammar Tags

Dependency	Tag	Word Grammar
Subject	SBJ	فاعل/ نائب فاعل/ مبتدأ/ اسم كان/ اسم ان/ اسم كاد/ اسم فاعل/ اسم مفعول
Object	OBJ	مفعول لفعل/ مفعول لاسم/ اسم مفعول/ مصدر / اسم مجرور
Predicate	PRD	خبر لمبتدأ / خبر ان/ خبر كان
Topic	TPC	مبتدأ
Idafa	IDF	مضاف اليه
Tamyiyz	TMZ	تمييز
Modifier	MOD	صفة/ حال/ ظرف
Flat	------	اسم علم/ اسم اعجمي / رقم/ علامات ترقيم مكررة

Table II: MADA+TOKAN POS Tags

POS tag	Tag abbreviation	Word Type
Verb	VRB	فعل معلوم
Passive Verb	VRB-pass	فعل مجهول
Nominal	NOM	اسم
Particle	PRT	حرف/ أداة
Punctuation	PNX	علامة ترقيم
Proper Noun	PROP	اسم علم
Error	ERR	خطأ
Unknown		غير معروف

Before starting applying the proposed approach, the following conditions should be met:

- Each tag for each POS is expressed using the following set (Word, Word Type, Level, POS Tag).
- Each requirement is a verbal sentence, and each verbal sentence is an action and for each action there is a subject and sometimes an object.
- The tag <PNX> tag means the end of the sentence by comma (,) or full stop (.), but in this phase sentences or user requirements statements will be separated not based on <PNX> tag - which represent (.), (،) in order to order to analyze verbal sentences, so when we say requirement or sentence, we mean a set of tags between two consecutive <VRB> tags.
- In UML terms, subjects called senders, objects called receivers, both subjects and objects called participants and actions called messages.

This approach is applied via two phases, the first phase including the scanning of the resulted tags from MADA+TOKAN to define the subjects set which is a subset of participants set. At the beginning of applying the proposed approach, the following sets should be declared:

Subjects= {}
Receivers = {}
Participants = {}

Subjects set is a set of distinct words that has a POS tag of <SBJ> in the results of parsing a specific scenario. *Subjects* set will be used in subsequent phases to find the callers of that scenario, *Receivers* is the set of the distinct receivers of the actions in the same scenario while the *Participants* is the set of distinct callers and receivers. *Subjects* set should be defined at the first phase of scanning the parsing results while *Receivers* and *Participants* sets will be defined and updated during the second phase of constructing sequence diagram key parts.

Also, a *sequence table* should be constructed and updated during the approach analysis process with the structure shown in Table III.

TABLE III. SEQUENCE DIAGRAM TABLE STRUCTURE

Statement #	Sender	Receiver	Message

B. Participants Identification

The participants of sequence diagram include:

- Sender/ Caller
- Main Actor
- Receiver

By applying the first phase of scanning MADA+TOKAN parsing results on ATM system scenario, we can find that the resulted Subjects are as follow:

Subjects = {البنك، الصراف، الزبون}

1) Sender Identification: The sender or the caller for each statement represents the subject of the action of that statement, which means senders are identified based on subject tags.

For each statement the sender of that statement is the subject of <SBJ > tag.

Example:

يقوم الزبون بإدخال البطاقة الى الصراف

Using MADA+TOKAN tool the statement parsing results as shown in Table IV.

Table IV: MADA+TOKAN POS Tags

Word	Word Type	Level	Grammar
يقوم	VRB	0	
1 الزبون	NOM	1	SBJ
+ ب	PRT	1	MOD
إدخال	NOM	3	OBJ
البطاقة	NOM	4	IDF
إلى	PRT	4	MOD
الصراف	NOM	6	OBJ
حيث	NOM	7	MOD

Here the main subject is <SBJ > tag which is <الزبون>

Generalization:

*Rule P1***:** For each user requirement statement with the following set of POS tags: <Word, NOM, level, VRB> <Word, NOM, level, SBJ> or <Word, NOM, level, SBJ> <Word, NOM, level, VRB> find the <Word, NOM, level,

SBJ> tag to find the sender or the caller of that statement. Then the founded subject should be added to the *Participants* group. If there is no <SBJ> tag, there is no sender and no message which means discarding the full statement.

2) *Main Actor Identification*: Main actor for any system is the first subject in that system, based on resulted tree bank of MADA+TOKAN tagger, the first subject should have the level number of (1) after a verb of level (0) which is the root.

Generalization:

Rule P2: To find the main actor, search for the subject <Word, NOM, 1, SBJ> tag of the level (1) in all resulted MADA+TOKAN POS tags. Then add it to *Participants* set, if it is not exist in it. If it is existing then just mark it as main actor.

3) *Receiver Identification*: The receiver for each statement represent the object for the action of that statement. So, receivers are identified based on objects tags. But, as the authors supposed, objects tags can be a receiver or a message. To find the receiver for each user requirement statement:

Find all < Word, NOM, level, OBJ > tags within each statement. And find the object that belongs to subjects group because each object in different point should be a subject (sender). If none of the objects are within subjects group then the last object of that statement is the receiver.

Update *Participants* set by adding the found receiver to it. Referring to Table IV, we can find that we have two objects in this statement:

OBJ = {<إدخال NOM 3 OBJ >-, <الصراف NOM 6 OBJ >-}

- The first object <إدخال > is not belonging to subjects group then it's not the receiver.
- The second object <الصراف> is belonging to subjects group then it is the receiver.
- Update Participants group and sequence table:

Generalization:

Rule P3: To find the receiver for each statement, apply the following rules on all <OBJ> tags within a statement:

For each <OBJ> tag of the following set < Word, Word- Type, level, OBJ >

Check if it the Word-Type is NOM then check if the object is belonging to Subjects group

If yes, then this object is a receiver Else, it is a message (in the next section)

Else, the last object in this statement is the receiver and all other objects are messages between the same sender and receiver within this statement.

Check if it the Word-Type is VRB then discard it

C. Message Identification

Messages are the actions for each statement, and usually the message is more than one tag, to find the message within user requirements statements, we have to find the <OBJ> tag that is not the receiver. Sometimes this tag is followed by an idafa <IDF> or modifier <MOD > tags to construct the message between two participants. Referring to Table IV, we can find that we have two objects in this statement:

- The first object <إدخال > is not belonging to subjects group then it is a message:
- The next tag for this tag is <البطاقة NOM 4 IDF > so the full message is <إدخال NOM 3 OBJ > <البطاقة NOM 4 IDF >, "إدخال البطاقة".
- The second object is <الصراف>, based on rule P2 it is a receiver and not a message.
- Update sequence table

Generalization:

Rule M1: To find the message for each statement, apply the following rules on all <OBJ> tags that are not within a subject statement:

If the object tags –are <Word, NOM, Level_NO, OBJ> and it is not belonging to the receivers group then it is a message and to find the message:

If the next tags is <Word, NOM, Level_NO, IDF> then the message is <Word, NOM, Level_NO, OBJ> + <Word, NOM, Level_NO, IDF>

Else if the next tag is <Word, NOM, Level_NO, MOD> then the message is <NOM, Level_NO, OBJ> + <Word, NOM, Level_NO, MOD>

Else the message is <Word, NOM, Level_NO, OBJ>

Update sequence table

D. Algorithm of Applying Heuristics

In this section the used algorithm for applying the proposed heuristics on the resulted tags from parsing user requirements in MADA+TOKAN is presented as follow:

Input: Arabic User Requirements
Result: Sequence Diagram
Subjects = {}, Receivers = {}, Participants = {}, Sequence_ table [ ] [ ]
Subjects= All <SBJ> tags
// Based on Subjects group
Find main actor based on Rule P2
Add main actor to Participants group and mark it as the initiator
// Each statement is a set of tags between two <VRB> tags
For all Arabic user requirements statements do
Apply Rule P1 to find the sender
Update Participants group and Sequence table Apply Rule P3 to find the receiver

Update Receivers, Participants groups and Sequence table
Apply Rule M1 to find the message Update Sequence table
end

By applying the algorithm on all statements of ATM system, the results will be as follows:

Participants = {{الزبون} * ،الصراف ،البنك}
Subjects = {الزبون، الصراف، البنك}
Receivers = {الزبون، الصراف، البنك}

While the final sequence table as shown in Table III. We can see that statement number 5 has been discarded based on Rule P1. The next step is transforming the results for each row in final sequence table (message, sender and receiver) into XMI to be drawn using UML drawing tools.

Table III: FINAL SEQUENCE TABLE

Statement #	Sender	Receiver	Message
1	الزبون	الصراف	إدخال البطاقة
2	الصراف	الزبون	طلب الرقم
3	الزبون	الصراف	الرقم السري
4	الصراف	البنك	المعلومات الى
5			
6	البنك	الصراف	ارسال النتيجة
7	الصراف	الزبون	عرض الخيارات
8	الزبون	الصراف	ادخال طلب
8	الزبون	الصراف	ه الى
9	البنك	الصراف	رسالة نجاح
10	الصراف	الزبون	ارسال اشعار

V. EVALUATION

The next step in this research is the evaluation of the proposed approach. Once the approach proves to be beneficial, it will be implemented as a software tool that can be used to generate the sequence diagram model from Arabic user requirements.

VI. CONCLUSION

In this paper, a new semi- automated approach for generating UML sequence diagrams from Arabic user requirements was proposed. The proposed approach is essential in object oriented applications, in requirements analysis phase and in software especially in generating UML sequence diagrams from Arabic user requirements. The proposed approach has the main advantage of dealing with Arabic language and also a set of heuristics were proposed and applied on a set of tokens resulted from natural language processing tool called MADA+TOKAN to obtain sequence diagram key parts which include (participants and messages). Finally, the proposed approach is to be validated and implemented in further research efforts.

VII. ACKNOLEDGEMENT

The authors would like to than the Software Engineering Research Group members at Palestine Polytechnic University, especially Mr. Ibrahim Nassar for his help regarding MADA+TOKAN tool and Dr. Khaled Daghmen for his help regarding software engineering and algorithms.

REFERENCES

[1] Gegentana. A Systematic Review of Automated Software Engineering. ADDIMaster of Science Thesis in Program Software Engineering and Management SON-WESLEY, 2011.

[2] Paul Harmon and Mark Watson. Understanding UML- The Developers Guide. Morgan Kaufmann Publishers, 2005.

[3] M. G. Ilieva and O. Ormandjieva. Models Derived from Automatically Analyzed Textual User Requirements . Software Engineering Research, Management and Applications, 2006. Fourth International Conference on, pages 13 –21, IEEE, 2006.

[4] Liwu Li. Translating use cases to sequence diagrams. Automated Software Engineering, 2000. Proceedings ASE 2000. The Fifteenth IEEE International Conference on, pages 293– 296, IEEE, 2000.

[5] Daniel Popescu, Spencer Rugaber, Nenad Medvidovic, and Daniel M. Berry. Reducing Ambiguities in Requirements Specifications Via Automatically Created Object-Oriented Models . Innovations for Requirement Analysis. From Stakeholders Needs to Formal Designs, 5320:103–124, Springer, 2008.

[6] Nayanamana Samarasinghe and Stephane S. Som. Generating a Domain Model from a Use Case Model . IASSE, page 278, 2005.

[7] Evaluation of Novel Approaches to Software Engineering (ENASE), 2014 International Conference on, IEEE, 978-989-758-065-9:19, 2014.

[8] P. A. J. Mason and S. Supsrisupachai. Paraphrasing use case descriptions and Sequence Diagrams: An approach with tool support. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2009. ECTI-CON 2009. 6th International Conference on, 2:722 – 725, IEEE, 2009.

[9] Jitendra Singh Thakur and Atul Gupta. Automatic generation of sequence diagram from use case specification . ISEC '14 Proceedings of the 7th India Software Engineering Conference, 20, ACM, 2014.

[10] Nabil Arman and Sari Jabbarin. Generating Use Case Models from Arabic User Requirements in a Semiautomated Approach Using a Natural Language Processing Tool. J. Intell. Syst, 24(2):277286, 2015.

[11] Ibrahim Nassar and Faisal Khamayseh. A Semi-Automated Generation of Activity Diagrams from Arabic User Requirements. NNGT Int. J. on Software Engineering, 2, 2015.

[12] Meryem Elallaoui, Khalid Nafil, and Raja Touahni. Automatic generation of UML sequence diagrams from user stories in Scrum process. 2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA), IEEE, page 16, 2015.

[13] Richa Sharma, Sarita Gulia, and K. K. Biswas. Automated generation of activity and sequence diagrams from natural language requirements.

[14] Bashar Nuseibeh and Steve Easterbrook. Requirements Engineering: A Roadmap . ACM, Proceedings of the Conference on The Future of Software Engineering, ICSE(ISBN:1-58113-253-0):35–46, 2000.

[15] Roger S. Pressman. IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, Std 830, 1998.

[16] Imran Sarwar Bajwa and M. Abbas Choudhary. FIntegrating natural language techniques in OO-Method. PSpringer-Verlag, 978-3-540-24523-0:560–571, 2005.

[17] Nizar Habash, Owen Rambow, and Ryan Roth. MADA+TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization. In Proceedings

of the fifth international conference on Language Resources and Evaluation, ISBN-13:978-0130969729:35–46, 2006.

[18] Tao Yue, Lionel C. Briand, and Yvan Labiche. Automatically Deriving UML Sequence Diagrams from Use Cases . Simula Research Laboratory, Technical Report 2010-04, 2010.